

# **Trabajo de Fin de Grado**

## Detección de Puntos Significativos en Imágenes para Reconstrucción 3D

Álvaro García Tenorio <sup>1</sup>  
**Doble Grado en Ingeniería Informática–Matemáticas**

**Universidad Complutense de Madrid**  
Facultad de Informática  
Departamento de Ingeniería del Software e Inteligencia Artificial

**Director:** Gonzalo Pajares Martinsanz  
**Codirector:** Antonio Valdés Morales

28 de mayo de 2019



<sup>1</sup>[alvgar14@ucm.es](mailto:alvgar14@ucm.es)



# Índice general

<b>1. Introducción</b>	<b>7</b>
1.1. Antecedentes . . . . .	7
1.2. Objetivos . . . . .	7
1.3. Plan de trabajo . . . . .	8
1.4. Organización de la memoria . . . . .	8
 <b>I Puntos de interés</b>	 <b>9</b>
<b>2. El operador SURF</b>	<b>11</b>
2.1. Detección de puntos de interés . . . . .	11
2.1.1. Convoluciones discretas . . . . .	11
2.1.2. Detector basado en el determinante Hessiano . . . . .	15
2.1.3. Espacio de escalas y supresión no máxima . . . . .	19
2.2. Descripción de puntos de interés . . . . .	20
2.2.1. Asignación de la orientación . . . . .	20
2.2.2. Construcción del descriptor del punto de interés . . . . .	22
 <b>3. Emparejamiento de puntos</b>	 <b>25</b>
3.1. Algoritmo de fuerza bruta . . . . .	25
3.2. Regla del cociente de Lowe y comprobación cruzada . . . . .	25
3.3. Resultados sobre imágenes no adaptadas . . . . .	26
 <b>4. Extensión afín</b>	 <b>27</b>
4.1. Idea del algoritmo . . . . .	27
4.2. Resultados . . . . .	28
 <b>II Reconstrucciones</b>	 <b>31</b>
<b>5. La cámara proyectiva finita</b>	<b>33</b>
5.1. Modelo geométrico . . . . .	33
5.1.1. Referencias de la retina . . . . .	33
5.1.2. Referencia conforme adaptada del espacio . . . . .	35
5.2. Matrices de cámara . . . . .	36
5.3. Matriz respecto de una referencia no adaptada . . . . .	37
 <b>6. Geometría epipolar</b>	 <b>39</b>
6.1. Introducción . . . . .	39
6.2. Obtención de la recta epipolar . . . . .	40
6.3. Matriz fundamental . . . . .	42
6.4. Homografía epipolar . . . . .	43
6.5. Ambigüedad proyectiva de la matriz fundamental . . . . .	44
6.6. Matrices de cámara y matriz fundamental . . . . .	46
6.7. Recapitulación . . . . .	47

<b>7. Reconstrucciones</b>	<b>49</b>
7.1. Cómputo de la matriz fundamental . . . . .	49
7.2. Teorema de reconstrucción proyectiva . . . . .	51
7.3. Cómputo de la triangulación . . . . .	51
7.4. Recapitulación . . . . .	52
 <b>III Resultados y conclusiones</b>	 <b>55</b>
<b>8. Resultados y conclusiones</b>	<b>57</b>
8.1. Resultados . . . . .	57
8.2. Conclusiones generales y trabajo futuro . . . . .	59
 <b>IV Anexos</b>	 <b>61</b>
<b>A. Descomposiciones de matrices</b>	<b>63</b>
A.1. Descomposición RQ de una matriz $3 \times 3$ . . . . .	63
A.2. Descomposición en valores singulares (SVD) . . . . .	64
<b>B. Geometría del punto unidad</b>	<b>65</b>
B.1. Caracterización del punto unidad . . . . .	65
B.2. Procedimiento de construcción . . . . .	65
<b>C. Código fuente</b>	<b>67</b>
<b>Índice general</b>	<b>71</b>



## Resumen

La realización de una reconstrucción tridimensional de una escena a partir de imágenes bidimensionales de la misma puede ser la aproximación escogida para resolver multitud de problemas.

En este documento se exponen una serie de procedimientos que, concatenados, producen una reconstrucción proyectiva de una escena 3D a partir de dos imágenes 2D de la misma.

Se comienza estudiando operadores para la detección y descripción de puntos de interés, así como técnicas de emparejamiento de descriptores. En una fase intermedia, se presenta el modelo de cámara proyectiva finita, junto con diversos resultados concernientes a las relaciones geométricas entre dos cámaras. Por último, se termina presentando el teorema de reconstrucción proyectiva, acompañado de algunos procedimientos auxiliares que permiten el cálculo robusto de reconstrucciones proyectivas.

Los resultados obtenidos son satisfactorios bajo los planteamientos formulados en el presente trabajo.

## Palabras clave

Puntos de interés, SURF, SIFT, ASIFT, Modelos de cámara, Geometría epipolar, RANSAC, Reconstrucciones 3D.

## Abstract

Getting a three-dimensional reconstruction of a given scene from two-dimensional images may be the chosen approximation for solving many interesting problems.

This document presents a set of procedures, that, once linked together, can be used to produce a projective reconstruction of a 3D scene from two 2D images.

First of all, some methods for detection and description of interest points are studied, followed by several descriptor matching techniques. The previous image-focused phase is followed by two camera-focused chapters about the finite projective camera model and epipolar geometry. Finally, the projective reconstruction theorem is stated and proved. Also, some complementary algorithms are presented for robustness and stability reasons.

## Keywords

Interest Points, SURF, SIFT, ASIFT, Camera Models, Epipolar Geometry, RANSAC, 3D Reconstructions.

## Licencia

Esta obra está sujeta a la licencia Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/4.0/>.





# Capítulo 1

## Introducción

### 1.1. Antecedentes

Hay numerosos problemas que pueden requerir de la realización de reconstrucciones tridimensionales para su resolución, y, en particular, bajo la restricción de ser desconocidos los parámetros intrínsecos de las cámaras involucradas en el proceso. Este es el caso, por ejemplo, de la navegación de vehículos autónomos.

En este trabajo se plantea el problema de reconstruir una escena 3D a través de dos imágenes bidimensionales de la misma. Esto se consigue mediante la aplicación de técnicas de reconstrucción proyectiva sobre las dos imágenes de la escena. Uno de los requisitos necesarios para la aplicación de estas técnicas consiste en establecer puntos de correspondencia entre las dos imágenes, que, en la terminología especializada se conocen como *puntos de interés*. Estos puntos necesitan ser lo más robustos posibles, desde el punto de vista de su detección y similitud entre las dos imágenes. Con ellos, y una vez establecido el modelo proyectivo de las cámaras es posible determinar la estructura 3D de la escena.

A día de hoy existen numerosas librerías bajo licencias de software libre que implementan gran cantidad de las herramientas necesarias para realizar reconstrucciones proyectivas, siendo quizá *OpenCV* [2] su máximo exponente, pues contiene tanto implementaciones de operadores para la detección y descripción de puntos de interés (SIFT [16], SURF [6],...) como métodos robustos basados en RANSAC [9] y descomposición en valores singulares que permiten estudiar las relaciones entre dos cámaras dentro de la misma escena [15].

### 1.2. Objetivos

El objetivo general del trabajo consiste en la detección de puntos significativos en más de una imagen para establecer correspondencias y determinar así la estructura de la escena 3D. Este objetivo general se desglosa en los siguientes objetivos específicos.

1. Exponer el fundamento teórico de los operadores SIFT/SURF, observando su comportamiento gracias a las implementaciones presentes en [2] y [22]. A esto se dedica esencialmente el capítulo 2.
2. Estudiar diversas técnicas de emparejamiento de descriptores de puntos de interés. Este es el cometido del capítulo 3.
3. Estudiar las limitaciones de los algoritmos SIFT/SURF así como exponer la extensión afín ASIFT [29]. Este trabajo se realiza en el capítulo 4.
4. Estudiar, desde un punto de vista geométrico, el modelo de cámara proyectiva finita, así como las relaciones existentes entre dos cámaras. Este objetivo se desarrolla durante los capítulos 5 y 6.

5. Enunciar y demostrar el teorema de reconstrucción proyectiva. Asimismo, exponer varios algoritmos que permitan realizar tareas esenciales en el proceso de obtención de reconstrucciones proyectivas de manera numéricamente estable y robusta a errores en el proceso de emparejamiento de puntos. Todo esto se realiza en el capítulo 7.

### 1.3. Plan de trabajo

El plan de trabajo seguido ha consistido en trabajar de forma más o menos independiente en los grupos de objetivos (1, 2, 3) y (4, 5). El tiempo invertido ha sido, aproximadamente, 30 semanas, siendo la distribución del mismo la siguiente.

- Estudio bibliográfico acerca los métodos SURF/SIFT: 3 semanas.
- Interpretación de los resultados obtenidos por el algoritmo SURF: 3 semanas.
- Redacción del capítulo 2 (incluidas revisiones): 1 semana.
- Interpretación de los resultados obtenidos por las técnicas de emparejamiento estudiadas y redacción del capítulo 3 (incluidas revisiones): 1 semana.
- Estudio bibliográfico del método ASIFT, interpretación de resultados y redacción del capítulo 4 (incluidas revisiones): 3 semanas.
- Estudio bibliográfico acerca del modelo de cámara proyectiva finita y redacción del capítulo 5 (incluidas revisiones): 6 semanas.
- Estudio bibliográfico sobre geometría epipolar y redacción del capítulo 6 (incluidas revisiones): 4 semanas.
- Estudio bibliográfico de algoritmos basados en RANSAC y descomposición en valores singulares para el cálculo de la matriz fundamental y la triangulación de puntos: 3 semanas.
- Redacción del capítulo 7 (incluidas revisiones): 3 semanas.
- Sucesivas revisiones menores: 3 semanas.

### 1.4. Organización de la memoria

La memoria se organiza en cuatro partes, quedando dedicada la parte I a los objetivos 1, 2 y 3, dejando a la parte II la resolución de los objetivos 4 y 5.

Las partes I y II pueden leerse de manera independiente, no obstante, dentro de cada parte, cada capítulo depende del anterior.

La parte III describe los resultados obtenidos, incluyendo una valoración de los mismos. Finalmente, la parte IV contiene los anexos que complementan diversos aspectos de la memoria.

#### Convenios de notación

En términos tipográficos, un vector columna será representado con una letra en negrita, por ejemplo  $\mathbf{v}$ . Siendo los vectores fila escritos como vectores columna traspuestos, por ejemplo,  $\mathbf{v}^\top$ . Por su parte, las matrices con más de una fila o más de una columna serán representadas con letras mayúsculas en tipografía de máquina de escribir, por ejemplo,  $\mathbf{A}$ . De esta manera queda claro a simple vista cuándo el texto se refiere a un elemento geométrico y cuándo se refiere a su representación en coordenadas.

Dados dos puntos proyectivos  $x$  e  $y$ , a lo largo del documento se denotará por  $xy$  a la recta que pasa por ambos.

#### Otras observaciones

Todas las figuras del documento han sido elaboradas gracias al software Geogebra [1]. Asimismo, todas las fotografías expuestas son de realización propia.

**Parte I**

**Puntos de interés**



## Capítulo 2

# El operador SURF

En este capítulo se describe de manera detallada el algoritmo SURF (**S**peeded-**U**p **R**obust **F**eatures) [6] para la detección y descripción de puntos de interés sobre imágenes. Asimismo, se observa el comportamiento del operador en función de algunos de sus parámetros configurables.

Durante todo este capítulo se toma como referencia principal la implementación del algoritmo disponible en la librería OpenCV [2].

El algoritmo SURF se basa en la misma idea que el operador SIFT (**S**cale **I**nvariant **F**eature **T**ransform) [16]. Lo que distingue a SURF son las aproximaciones que aplica para acelerar los cálculos que realiza SIFT, reduciendo el tiempo de ejecución sin un perjuicio apreciable en los resultados finales.

### 2.1. Detección de puntos de interés

El primer paso para realizar una reconstrucción tridimensional de una escena a partir de imágenes de la misma, es encontrar correspondencias entre dichas imágenes.

Para obtener dichas correspondencias es necesario detectar *puntos de interés*, estos son, informalmente, puntos seleccionados sobre zonas notables de la imagen, como, por ejemplo, esquinas. Dichos puntos, además, deben ser robustos a diversas perturbaciones, por ejemplo, la escala o las rotaciones. A continuación se exponen los conceptos teóricos encaminados a la detección de puntos de interés.

#### 2.1.1. Convoluciones discretas

Antes de comenzar a describir el algoritmo per sé, es necesario definir qué es una *convolución*. Informalmente, las convoluciones modifican el contenido de una imagen mediante operaciones de vecindad con un núcleo predefinido, siendo el filtrado un claro ejemplo de ellas.

Así pues, en primer lugar, es necesario definir el concepto de imagen.

**Definición 2.1.1** (Imagen). Una *imagen* es una función  $p : \mathbb{Z}^2 \rightarrow \mathbb{R}$ . Usualmente se denota a una imagen por  $p$ , por ser la inicial de *photo*.

Nótese que, por simplicidad, se está suponiendo que la imagen no tiene bordes, hipótesis que será refinada más adelante en esta subsección.

El hecho de que una imagen quede representada por una función  $\mathbb{Z}^2 \rightarrow \mathbb{R}$  es razonable si se piensa que los pares  $(x, y) \in \mathbb{Z}^2$  representan los píxeles de la imagen, mientras que  $p(x, y)$  representa la *intensidad* del píxel  $(x, y)$ .

Es conveniente hablar de la *intensidad* de un píxel, puesto que SURF, al igual que muchos otros algoritmos similares, únicamente trabaja con imágenes en escala de grises. En la práctica, se suele usar una representación de 8 bits por píxel, por lo que se tiene que  $p(x, y) \in [0, 255] \cap \mathbb{Z}$ .

**Definición 2.1.2.** Dadas dos funciones  $p : \mathbb{Z}^2 \rightarrow \mathbb{R}$  y  $f : S \subset \mathbb{Z}^+ \times \mathbb{Z}^+ \rightarrow \mathbb{R}$ , con  $S$  finito, definimos su convolución en el punto  $(x, y)$  como

$$(f * p)(x, y) := \sum_{(i,j) \in S} f(i, j)p(x - i, y - j) \quad (2.1)$$

La función  $p$  representa la imagen a tratar por el algoritmo, mientras que la función  $f$  (llamada así por ser la inicial de *filter*) representa el filtro que actúa sobre la imagen. A las funciones que actúan como filtros también se las denomina “núcleos”.

Nótese que la convolución de una imagen con un núcleo es, a su vez, una imagen, a la que llamaremos *imagen convolucionada*.

La figura 2.1 ayuda a visualizar el proceso de convolución de una imagen con un filtro cuadrado. Podemos apreciar que la esquina inferior derecha del filtro se encuentra sobre el píxel  $(x, y)$  de la imagen original, lo cual quizá no sea del todo conveniente, ya que el píxel  $(x, y)$  de la imagen convolucionada no depende de todo el vecindario del píxel  $(x, y)$  en la imagen original, sino únicamente de lo que se encuentra arriba y a su izquierda.

$f(2, 2)$	$f(2, 1)$	$f(2, 0)$	
$f(2, 1)$	$f(1, 1)$	$f(1, 0)$	
$f(0, 2)$	$f(0, 1)$	$p(x, y)$ $f(0, 0)$	

Figura 2.1: Ilustración de una imagen (gris) con un filtro (naranja) superpuesto.

Para solventar esta problemática, el algoritmo SURF, siempre usa filtros cuadrados cuyo lado tiene un número impar de píxeles. Esto se hace para garantizar que el filtro siempre tiene un píxel central. Formalmente, los filtros que usaremos serán de la forma  $f : ([0, 2n] \times [0, 2n]) \cap \mathbb{Z}^2 \rightarrow \mathbb{R}$ , siendo el píxel central  $(n, n)$ .

Dicho esto, en la práctica, en vez de usar la definición 2.1.2 para el cálculo de la convolución, se usará la siguiente fórmula alternativa.

$$(f \star p)(x, y) := (f * p)(x + n, y + n) \quad (2.2)$$

	$f(2, 2)$	$f(2, 1)$	$f(2, 0)$
	$f(1, 2)$	$p(x, y)$ $f(1, 1)$	$f(1, 0)$
	$f(0, 2)$	$f(0, 1)$	$p(x+1, y+1)$ $f(0, 0)$

Figura 2.2: Ilustración del cálculo de  $(f \star p)(x, y)$ .



### Convolución sobre los bordes

Hasta ahora se había supuesto que las imágenes con las que trataríamos estarían definidas sobre todo par  $(x, y)$ , esto en la práctica no es así. Este hecho es problemático, por ejemplo, cuando se quiere aplicar un filtro sobre un píxel que se encuentra cerca del borde de la imagen, pues puede haber píxeles alrededor de dicho punto que no tengan un valor definido.

Existen diversas estrategias para definir las intensidades de estos píxeles ausentes, de manera que el resultado de la convolución no se vea muy alterado en los bordes. La más utilizada en las implementaciones de SURF, en particular, en la propuesta por [22], es la de realizar una “ampliación especular” de la imagen.

Esta ampliación se puede calcular únicamente cuando sea necesario, sin embargo, dada la frecuencia con la que SURF realiza convoluciones, es conveniente calcularla al principio, a modo de preprocesamiento de la imagen.

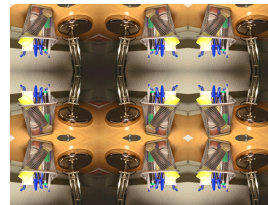
Si  $p$  está definida sobre  $[0, M] \times [0, N] \cap \mathbb{Z}^2$ , es sencillo deducir las ecuaciones que nos permiten ampliar especularmente la imagen

$$\left. \begin{array}{l} p(i, N+d) := p(i, N-d-1) \\ p(i, -d) := p(i, d-1) \\ p(M+d, j) := p(M-d-1, j) \\ p(-d, j) := p(d-1, j) \\ p(-d, N+h) := p(d-1, N-h-1) \\ p(-d, -h) := p(d-1, h-1) \\ p(M+d, N+h) := p(M-d-1, N-h-1) \\ p(M+d, -h) := p(M-d-1, h-1) \end{array} \right\} \begin{array}{l} 0 \leq i \leq M-1, 0 \leq d \leq N-1 \\ 0 \leq j \leq N-1, 0 \leq d \leq M-1 \\ 0 \leq d \leq M-1, 0 \leq h \leq N-1 \end{array} \quad (2.3)$$

Como se verá más adelante, antes de ejecutar el algoritmo en sí mismo, es posible conocer la dimensión del mayor filtro a aplicar, digamos  $2n+1$ , de esta forma, como mucho se necesitará una ampliación especular de  $n$  píxeles de anchura, y se puede asumir como hipótesis que  $n \leq \min(M-1, N-1)$ , de modo que las ecuaciones (2.3) siguen pudiéndose aplicar. La figura 2.3 ilustra los efectos de una ampliación especular.



(a) Imagen sin tratar



(b) Imagen ampliada

Figura 2.3: Comparación entre una imagen sin tratar y su ampliación especular. La ampliación especular ha sido generada con el archivo “[mirror.py](#)”

### Imágenes integrales

El algoritmo SURF realiza gran cantidad de convoluciones, por lo que resulta muy conveniente realizar algún acondicionamiento sobre la imagen a tratar que después acelere el proceso de convolución.

El cálculo previo más importante a realizar antes de comenzar a ejecutar el algoritmo es el del cálculo de la *imagen integral* de  $p$ .

**Definición 2.1.3** (Imagen integral). Dada una imagen  $p$ , se define su imagen integral  $\Gamma$  como

$$\Gamma(x, y) := \sum_{i=0}^x \left( \sum_{j=0}^y p(i, j) \right) \quad (2.4)$$

La figura 2.5 muestra una imagen junto con su imagen integral asociada.

Si la imagen original tiene  $NM$  píxeles, el coste de calcular la imagen integral es  $\mathcal{O}(NM)$  si se aplica el algoritmo 1, que es el propuesto en la implementación adjunta a [22]. Existen otros algoritmos para el cómputo de imágenes integrales que explotan el paralelismo, por ejemplo, los propuestos en [7].

---

**Algoritmo 1** Cálculo de la imagen integral
 

---

**Entrada:** Imagen inicial  $p$

**Salida:** Imagen integral  $\Gamma$  de  $p$

```

    {Calculamos la imagen integral de la primera fila}
1:  $\Gamma(0, 0) = p(0, 0)$ 
2: para  $j \in \{1, \dots, N - 1\}$  hacer
3:    $\Gamma(0, j) = p(0, j) + \Gamma(0, j - 1)$ 
4: fin para
    {Calculamos el resto de la imagen integral}
5: para  $i \in \{1, \dots, M - 1\}$  hacer
6:    $\xi = 0$ 
7:   para  $j \in \{0, \dots, N - 1\}$  hacer
8:      $\xi = \xi + p(i, j)$ 
9:      $\Gamma(i, j) = \xi + \Gamma(i - 1, j)$ 
10:  fin para
11: fin para
12: devolver  $\Gamma$ 
  
```

---

Una propiedad fundamental de las imágenes integrales es que, una vez obtenidas, es posible calcular la suma de las intensidades de los píxeles de cualquier subrectángulo de la imagen original  $p$  en tiempo constante (únicamente con la ayuda de tres operaciones aritméticas y cuatro accesos a memoria).

En efecto, para calcular la suma de la intensidad de todos los píxeles dentro del subrectángulo de  $p$  determinado por los píxeles distintos  $(i_1, j_1)$  e  $(i_2, j_2)$ , de los que únicamente sabemos que  $i_1 \leq i_2$  y  $j_1 \leq j_2$ , basta con usar la igualdad

$$\sum_{i=i_1}^{i_2} \left( \sum_{j=j_1}^{j_2} p(i, j) \right) = \Gamma(i_2, j_2) - \Gamma(i_2, j_1) - \Gamma(i_1, j_2) + \Gamma(i_1, j_1) \quad (2.5)$$

La demostración de la misma es inmediata. La figura 2.4 ilustra gráficamente este concepto.

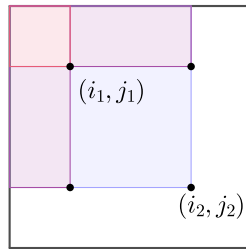


Figura 2.4: Ilustración de la igualdad (2.5)

Como se expone en la siguiente sección, el uso de imágenes integrales es especialmente fructífero a la hora de acelerar el cálculo de convoluciones con filtros rectangulares constantes de tamaño grande, ya que basta con calcular la suma de intensidades del rectángulo en la imagen original (coste constante) y multiplicar dicho valor por la constante del filtro.

Es por este motivo que SURF trata de simplificar al máximo los filtros que usa, de manera que su aplicación sobre un píxel se realice en tiempo constante independientemente del tamaño del filtro.

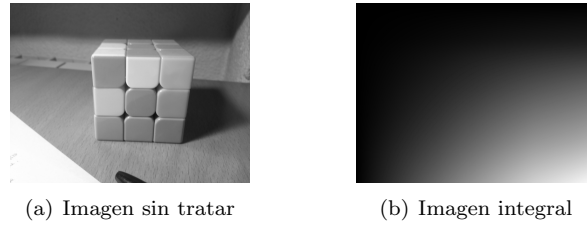


Figura 2.5: Comparación entre una imagen sin tratar y su imagen integral. La imagen integral ha sido generada con el archivo “[integral.py](#)”, que implementa el algoritmo 1.

### 2.1.2. Detector basado en el determinante Hessiano

Por el momento, será conveniente pensar en las imágenes, no como funciones discretas, como hasta ahora, sino como funciones continuas  $\mathbb{R}^2 \rightarrow \mathbb{R}$ , y, por tanto, como grafos.

Para comenzar, se recuerda que, dada una matriz  $\mathbf{A}$ , cuadrada de orden  $n$ , su determinante es el producto de sus autovalores. Es decir

$$\det(\mathbf{A}) := \prod_{i=1}^n \lambda_i, \text{ siendo } \lambda_i \text{ los autovalores de } \mathbf{A} \quad (2.6)$$

Se puede asumir como hipótesis que las segundas derivadas de la imagen son continuas y, por tanto, por el teorema de Schwartz, la matriz Hessiana es simétrica en todo punto, de lo que se desprende, tras aplicar el teorema espectral, que todos sus autovalores son reales. En concreto, al ser una matriz  $2 \times 2$ , posee dos autovalores.

Además, es conveniente interpretar a la matriz Hessiana como una forma cuadrática.

**Definición 2.1.4.** Se dice que una función  $p : \mathbb{R}^n \rightarrow \mathbb{R}$  diferenciable, es **convexa** en  $x$  si hay un  $r \in \mathbb{R}$  de manera que para todo  $y$  tal que  $\|x - y\| \leq r$ , se cumple que

$$p(y) \geq p(x) + \nabla p(x)^\top (y - x) \quad (2.7)$$

Análogamente, si  $p(y) \leq p(x) + \nabla p(x)^\top (y - x)$  se dirá que  $p$  es **cóncava** en  $x$ .

Nótese que el miembro derecho de la ecuación (2.7) coincide con la ecuación del plano tangente a  $p$  en  $p(x)$ . Por ende, si  $p$  es convexa en  $x$ , hay un disco alrededor de  $x$  donde se cumple que la función queda por encima del plano tangente. La figura 2.6 muestra la situación en el caso cóncavo.

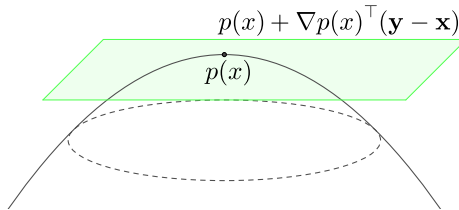


Figura 2.6: Ilustración de una función cóncava en  $x$ , y su plano tangente en  $p(x)$ .

**Teorema 2.1.1.** Dada una función  $p : \mathbb{R}^2 \rightarrow \mathbb{R}$  y un punto  $x \in \mathbb{R}^2$  tal que los autovalores de la matriz Hessiana de  $p$  en  $x$  son positivos, entonces  $p$  es estrictamente convexa en  $x$ .

*Demostración.* Por el teorema de Taylor se tiene que

$$p(y) = p(x) + \nabla p(x)^\top (y - x) + \frac{1}{2} (y - x)^\top \mathbf{H}_{\lambda x + (1-\lambda)y} (y - x) \quad (2.8)$$

Para cierto  $\lambda \in [0, 1]$ , representando  $\mathbf{H}_{\lambda x + (1-\lambda)y}$  la matriz Hessiana evaluada en  $\lambda x + (1 - \lambda)y$ .

Como la función que a cada matriz Hessiana le asigna su par de autovalores<sup>1</sup> es continua, y los autovalores de  $H_x$  son positivos, hay un  $r \in \mathbb{R}$  de manera que los autovalores de  $H_y$  son positivos, para todo  $y$  tal que  $\|x - y\| < r$ .

Por tanto, si  $\|x - y\| < r$ , los autovalores de  $H_{\lambda x + (1-\lambda)y}$  son positivos, en consecuencia,  $H_{\lambda x + (1-\lambda)y}$  es una forma cuadrática definida positiva, por ende  $p(y) > p(x) + \nabla p(x)^\top (y - x)$ , que es la definición de convexidad estricta en  $x$ . ■

**Corolario 2.1.2.** *Dada una función  $p : \mathbb{R}^2 \rightarrow \mathbb{R}$  y un punto  $x \in \mathbb{R}^2$  tal que los autovalores de la matriz Hessiana de  $p$  en  $x$  son negativos, entonces  $p$  es cóncava en  $x$ .*

La idea subyacente detrás del operador SURF es calcular el determinante Hessiano de cada punto de la imagen, seleccionando aquellos puntos cuyo determinante supere cierto umbral positivo  $T$ .

Como el determinante Hessiano es el producto de los autovalores de la matriz Hessiana, dichos autovalores asociados a los puntos seleccionados por SURF serán, o bien ambos positivos, o bien ambos negativos.

En consecuencia, SURF selecciona puntos donde la imagen tiene una convexidad (o concavidad) muy pronunciada.

### Cómo son los puntos detectados por SURF

Para tener una intuición más sólida acerca de cómo son los puntos de interés que detecta SURF, es conveniente estudiar cómo varía la intensidad de la imagen alrededor de dichos puntos.

Si  $x$  es un punto de interés, como  $H_x$  es diagonalizable, hay un cambio de referencia ortogonal de manera que las nuevas coordenadas de  $x$  son  $(0,0)$  y la matriz  $H_x$  pasa a ser diagonal, siendo los elementos de su diagonal sus autovalores. De ahora en adelante se supondrá que dicho cambio de referencia ha sido realizado.

Por el teorema de Taylor, se verifica la ecuación (2.8), y, como  $p$  es continua, para puntos cercanos  $x$  se cumple que

$$p(y) \approx p(x) + \nabla p(x)^\top \mathbf{y} + \frac{1}{2} \mathbf{y}^\top H_x \mathbf{y} \quad (2.9)$$

Hecho esto, conviene escribir  $\mathbf{y} = r (\cos \theta \quad \sin \theta)^\top$ , luego, sustituyendo en la ecuación (2.9)

$$p(y) \approx p(x) + \nabla p(x)^\top \mathbf{y} + \frac{1}{2} r^2 (\lambda_1 \cos^2 \theta + \lambda_2 \sin^2 \theta) \quad (2.10)$$

Como los autovalores de  $H_x$  tienen el mismo signo, se tiene dos situaciones de especial interés.

- Los autovalores  $\lambda_1$  y  $\lambda_2$  de  $H_x$  son similares. En tal caso, si  $\lambda \approx \lambda_1 \approx \lambda_2$  se tiene que

$$p(y) \approx p(x) + \nabla p(x)^\top \mathbf{y} + \frac{1}{2} r^2 \lambda \quad (2.11)$$

Por ende, salvando el término  $\nabla p(x)^\top \mathbf{y}$ , el valor de  $p(y)$ , siendo  $y$  un punto sobre una circunferencia de radio  $r$  con centro en  $x$ , es independiente de su posición dentro de la circunferencia. Esta situación queda ilustrada en la figura 2.6.

A estos puntos de interés se les suele llamar **blobs**.

- Si un autovalor  $\Lambda$  es mucho más grande que el otro se tiene que

$$p(y) \approx p(x) + \nabla p(x)^\top \mathbf{y} + \frac{1}{2} r^2 \Lambda \cos^2 \theta \quad (2.12)$$

Luego, observando la primera aproximación no lineal, el valor de  $p(y)$  aumenta según  $\theta$  se aproxima a 0 o a  $\pi$ , es decir, cuanto más cercano esté  $y$  del eje correspondiente al autovalor  $\Lambda$ , mayor será su valor. Esta situación queda ilustrada en la figura 2.7.

Por ende, SURF, además de **blobs**, detecta esquinas y bordes.

La figura 2.8 ilustra la detección de **blobs** por parte de SURF (pomos) y la detección de bordes y esquinas (divisiones entre cajones).

<sup>1</sup>Esta función está bien definida, pues los autovalores son las raíces de un polinomio de grado 2 existiendo para su cálculo una fórmula explícita.

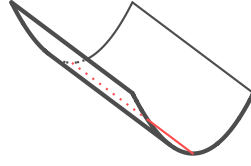
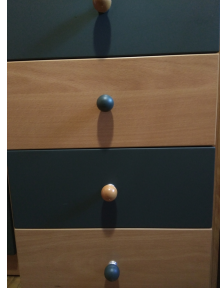
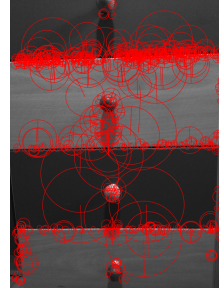


Figura 2.7: Ilustración del entorno de un punto de interés cuya diferencia de autovalores es grande.



(a) Imagen sin tratar



(b) Puntos de interés

Figura 2.8: Ilustración (en rojo) de los puntos de interés detectados por SURF.

### Convoluciones y derivadas parciales

Llegados a este punto, es conveniente volver a pensar en las imágenes como funciones discretas.

Aunque calcular el determinante de la Hessiana discreta de la imagen en cada punto sería posible, no es razonable, puesto que las imágenes reales no tienen tanta regularidad como se ha supuesto hasta ahora, con lo que podríamos obtener resultados poco fiables.

Una estrategia habitual en estos casos es aplicar primero un filtro de pequeño tamaño a la imagen original. Dicho filtro suele ser una aproximación de la aplicación Gaussiana definida en la ecuación (2.13).

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.13)$$

No obstante, en la práctica tampoco se hace esto, ya que se puede demostrar que las convoluciones se comportan bien con las derivadas, tal y como muestra la ecuación (2.14) (siendo este resultado también válido para el caso discreto)

$$\frac{\partial}{\partial_{xx}}(f * p) = \left( \frac{\partial}{\partial_{xx}} f \right) * p \quad (2.14)$$

y, análogamente, para el resto de derivadas parciales.

La consecuencia principal de esto es que se verifica (también en el caso discreto) que

$$H(f * p) = \begin{pmatrix} \frac{\partial}{\partial_{xx}} f * p & \frac{\partial}{\partial_{xy}} f * p \\ \frac{\partial}{\partial_{xy}} f * p & \frac{\partial}{\partial_{yy}} f * p \end{pmatrix} \quad (2.15)$$

Por ende, en lugar de tener que calcular derivadas discretas, se pueden calcular a priori las derivadas discretas del filtro a aplicar, que siempre es el mismo.

Dicho esto, la idea de SURF consiste en la aplicación de 3 filtros (que simularán las derivadas segundas de la gaussiana) sobre cada píxel.

Con la aplicación de los tres filtros, se consiguen los tres valores que determinan la matriz Hessiana en dicho píxel (pues es una matriz simétrica) y, por tanto, el determinante de la misma. En la subsección 2.1.3 se expone dónde y cómo se almacenan estos valores.

### Aproximación con “filtros caja”

La diferencia de SURF respecto de otros algoritmos como SIFT, es que realiza una aproximación extrema de estas derivadas (figura 2.9) a base de “filtros caja”, lo cual permite hacer uso de las

imágenes integrales, pudiendo realizar convoluciones de tamaño arbitrario en tiempo constante.

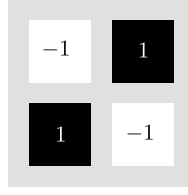


Figura 2.9: Filtro  $9 \times 9$  usado para aproximar la derivada segunda cruzada de la gaussiana.

A continuación se expone por qué es razonable aproximar las derivadas segundas de la función gaussiana mediante este tipo de filtros. Para ello, se presentan las conclusiones de un estudio detallado de los extremos de dichas derivadas. Dicho estudio puede consultarse en el archivo de *SageMath* [4] “[Gaussiana.ipynb](#)”.

Las derivadas segundas de la gaussiana son

$$\begin{aligned} g_{xx}(x, y, \sigma) &= -\frac{e\left(-\frac{x^2+y^2}{2\sigma^2}\right)}{2\pi\sigma^4} + \frac{x^2 e\left(-\frac{x^2+y^2}{2\sigma^2}\right)}{2\pi\sigma^6} \\ g_{yy}(x, y, \sigma) &= -\frac{e\left(-\frac{x^2+y^2}{2\sigma^2}\right)}{2\pi\sigma^4} + \frac{y^2 e\left(-\frac{x^2+y^2}{2\sigma^2}\right)}{2\pi\sigma^6} \\ g_{xy}(x, y, \sigma) &= \frac{xy e\left(-\frac{x^2+y^2}{2\sigma^2}\right)}{2\pi\sigma^6} \end{aligned} \quad (2.16)$$

Para estudiar los extremos de estas funciones, en primer lugar, es necesario calcular sus puntos críticos, es decir, los puntos que anulan sus gradientes, según la siguiente expresión

$$\begin{aligned} \{(x, y) \in \mathbb{R}^2 : \nabla g_{xx}(x, y) = 0\} &= \{(0, 0), (-\sqrt{3}\sigma, 0), (\sqrt{3}\sigma, 0)\} \\ \{(x, y) \in \mathbb{R}^2 : \nabla g_{yy}(x, y) = 0\} &= \{(0, 0), (0, -\sqrt{3}\sigma), (0, \sqrt{3}\sigma)\} \\ \{(x, y) \in \mathbb{R}^2 : \nabla g_{xy}(x, y) = 0\} &= \{(0, 0), (-\sigma, -\sigma), (\sigma, -\sigma), (-\sigma, \sigma), (\sigma, \sigma)\} \end{aligned} \quad (2.17)$$

Para dilucidar si estos puntos son máximos, mínimos o puntos de silla, se necesitan conocer las matrices Hessianas de las derivadas segundas de la gaussiana evaluadas en cada uno de sus puntos críticos. A modo de ejemplo, se presentan dichas matrices para  $g_{xx}$ .

$$\begin{aligned} H_{xx}(0, 0) &= \begin{pmatrix} \frac{3}{2\pi\sigma^6} & 0 \\ 0 & \frac{1}{2\pi\sigma^6} \end{pmatrix} \\ H_{xx}(\sqrt{3}\sigma, 0) = H_{xx}(-\sqrt{3}\sigma, 0) &= \begin{pmatrix} -\frac{3e\left(-\frac{3}{2}\right)}{\pi\sigma^6} & 0 \\ 0 & -\frac{e\left(-\frac{3}{2}\right)}{\pi\sigma^6} \end{pmatrix} \end{aligned} \quad (2.18)$$

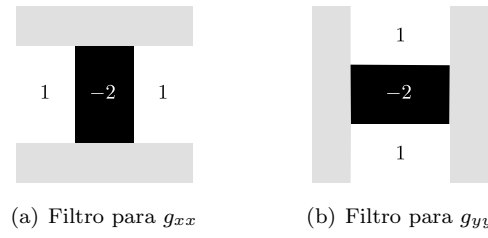
De la ecuación 2.18 se deduce que  $(0, 0)$  es un mínimo de  $g_{xx}$ , mientras que  $(-\sqrt{3}\sigma, 0)$  y  $(\sqrt{3}\sigma, 0)$  son máximos. Los valores de  $g_{xx}$  sobre estos puntos son

$$\begin{aligned} g_{xx}(0, 0) &= -\frac{1}{2\pi\sigma^4} \approx -0,0767 \\ g_{xx}(-\sqrt{3}\sigma, 0) = g_{xx}(\sqrt{3}\sigma, 0) &= \frac{e\left(-\frac{3}{2}\right)}{\pi\sigma^4} \approx 0,0342 \end{aligned} \quad (2.19)$$

Las aproximaciones de la ecuación 2.19 han sido realizadas suponiendo que  $\sigma = 1, 2$ , tal y como se indica en [6]. No obstante, SURF aproxima  $g_{xx}(0, 0)$  por  $-2$ , mientras que  $g_{xx}(-\sqrt{3}\sigma, 0)$  lo aproxima por 1. Es interesante notar que no hay ningún  $\sigma$  para el cual se verifiquen estas aproximaciones.

Del estudio de los extremos de  $g_{yy}$  se concluye que  $(0, 0)$  es un mínimo, mientras que  $(0, -\sqrt{3}\sigma)$  y  $(0, \sqrt{3}\sigma)$  son máximos. Sus valores son idénticos a los de  $g_{xx}$ , así como las aproximaciones realizadas por SURF.

Por último, del estudio de los extremos de  $g_{xy}$  se obtiene que  $(-\sigma, \sigma)$  y  $(\sigma, -\sigma)$  son mínimos, mientras que  $(\sigma, \sigma)$  y  $(-\sigma, -\sigma)$  son máximos, siendo  $(0, 0)$  un punto de silla.

Figura 2.10: Filtros  $9 \times 9$  usados para aproximar  $g_{xx}$  y  $g_{yy}$ 

### 2.1.3. Espacio de escalas y supresión no máxima

Una de las propiedades más deseables en un operador para la detección de puntos de interés es la “invarianza por escalas”.

Para conseguir este objetivo, es habitual que los algoritmos de detección de puntos de interés simulen mediante diversas técnicas, acercamientos o alejamientos de la imagen dada, para así obtener un análisis de la misma “a varias escalas”. A este proceso se le denomina *construcción del espacio de escalas*.

En el caso de SURF, esta simulación de las escalas se realiza procesando la imagen (cálculo del determinante Hessiano de sus píxeles) varias veces con un tamaño de filtro diferente cada vez.

#### Organización del espacio de escalas en SURF

SURF organiza su “espacio de escalas” en diferentes secciones llamadas *octavas*. Para cada octava se procesará la imagen cierto número fijo de veces, cada vez con filtros de mayor tamaño.

Llamaremos  $s_{i,k}$  al lado del filtro aplicado en la iteración  $i$ -ésima de la octava  $k$ -ésima. Los tamaños de estos filtros vienen dados por las siguientes ecuaciones.

$$\begin{aligned} s_{1,1} &:= 9 \\ s_{1,k} &:= s_{2,k-1} \\ s_{i,k} &:= 2^k 3 + s_{i-1,k} \end{aligned} \tag{2.20}$$

De esta manera, los tamaños de filtro a aplicar en la octava 1 serán: 9, 15, 21, 27, ..., mientras que los de la octava 2 serán: 15, 27, 39, 51, ... Los filtros aplicados serán versiones reescaladas de los filtros caja que aproximan las derivadas segundas de la función gaussiana (ver figura 2.9).

Nótese que, las octavas “se solapan” en el sentido de que en principio habría que volver a procesar la imagen con un tamaño de filtro ya aplicado anteriormente. Esta redundancia es necesaria, tal y como se explica en el siguiente apartado. Al contrario de lo que se pueda pensar, esto no supone un aumento en el coste del algoritmo, ya que, simplemente, se hará referencia a los resultados previamente computados.

Además de aplicar filtros de tamaño cada vez mayor, muchas implementaciones no calculan en cada iteración el determinante Hessiano de todos los píxeles, sino que recorren la imagen con paso  $2^{k-1}$  y calculan el Hessiano únicamente en los píxeles recorridos. Esto acelera el algoritmo y, en general, no supone una pérdida apreciable en la efectividad del mismo.

Tras el procesado de la imagen, cada píxel tendrá asignado un conjunto de valores, que se corresponderá con los determinantes hessianos calculados para cada octava y cada iteración dentro de la octava.

#### Selección de puntos de interés

Tras realizar el procesado de la imagen, se deben seleccionar los puntos que se consideran “de interés”. Para ello, se aplica un procedimiento de *supresión no máxima*, que se presenta de forma esquemática en la figura 2.11.

1. Dada una octava y un número de iteración dentro de dicha octava, de manera que esta iteración no sea ni la primera ni la última, se dividirá la imagen en bloques de píxeles de tamaño  $3 \times 3$ .

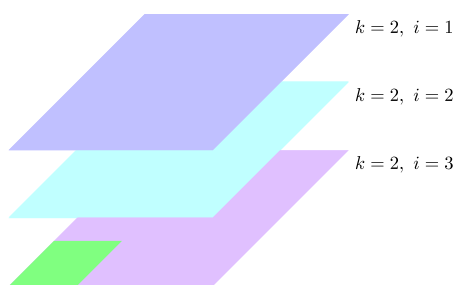
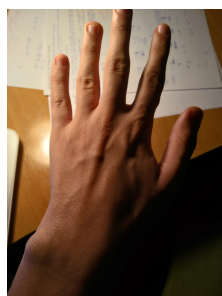


Figura 2.11: Ilustración de un bloque generado para la octava 2 en su segunda iteración. El píxel verde es el de determinante máximo.

Se exige además que los 9 píxeles del bloque hayan sido procesados para dicha octava.

2. Se transforma cada bloque en un cubo  $3 \times 3 \times 3$ , mediante la adición de los bloques correspondientes a las iteraciones anterior y posterior dentro de dicho octava.
3. Finalmente, se selecciona el píxel con mayor determinante (asociado a la octava y a la iteración especificadas) de cada cubo, siempre y cuando supere cierto umbral  $T$  definido por el usuario. Dicho píxel, junto con su número de octava e iteración es considerado un punto de interés.



(a) Imagen sin tratar



(b) Puntos de interés

Figura 2.12: Ilustración (en rojo) de los puntos de interés detectados por SURF. El radio de la circunferencia representa la escala a la que fue detectado, mientras que la dirección de la recta representa la orientación del punto.

## 2.2. Descripción de puntos de interés

La necesidad de la construcción de descriptores viene dada por la situación en la que disponemos de dos imágenes del mismo objeto. En estas circunstancias usualmente interesa establecer una correspondencia entre los puntos de interés de cada imagen, por ejemplo, para realizar una reconstrucción tridimensional del objeto.

Para establecer dicha correspondencia, se acompaña a cada punto de interés de información adicional (propiedades), a la que se llama *descriptor*. Dicha información adicional debe tener la propiedad de ser similar para puntos de interés correspondientes.

A continuación se detalla cómo SURF construye los descriptores de los puntos de interés.

### 2.2.1. Asignación de la orientación

Tras la fase de detección de puntos de interés, se dispone de una lista de ternas que contienen la posición del punto, la octava en la que se detectó y la iteración concreta dentro de dicha octava.



A partir de esta información se puede calcular, mediante las ecuaciones (2.20), el tamaño del filtro  $s$  con el que se calculó el determinante del punto.

Una vez obtenido  $s$ , SURF traza un círculo de radio  $6s$  alrededor del punto de interés. Dentro de dicho círculo se seleccionan puntos equiespaciados con paso  $s$  (los píxeles naranjas en la figura 2.13). Sobre cada uno de los puntos seleccionados se aplica el procedimiento descrito en el siguiente apartado.

### Cálculo de las derivadas parciales discretas

1. Sobre cada punto seleccionado se aplican dos **filtros de Haar** cuadrados de tamaño  $4s + 1$  (uno horizontal y otro vertical) para el cálculo de las derivadas parciales vertical y horizontal, respectivamente.

En la figura 2.13 se muestra uno de estos filtros.

2. Una vez calculadas las derivadas discretas, a las que llamaremos  $D_x$  y  $D_y$ , almacenaremos el par  $(D_x, D_y)g(x, y)$ , siendo  $g$  una gaussiana centrada en el punto de interés con  $\sigma = 2s$  evaluada en el punto actual  $(x, y)$ .

La idea de ponderar los resultados de esta manera es que las derivadas de puntos alejados del punto de interés tengan menos peso en la decisión de la orientación que las derivadas de puntos más cercanos.

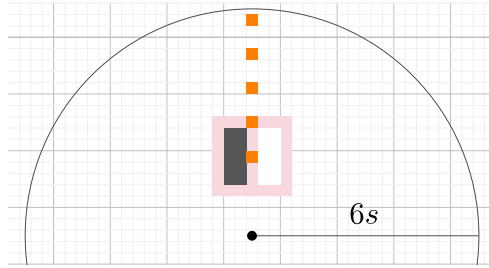


Figura 2.13: Ilustración no a escala. Sobre el punto más bajo ilustrado, se muestra el filtro de Haar que se usará para calcular la derivada parcial horizontal sobre dicho punto. La parte negra del filtro tendrá un valor de  $-1$ , mientras que la parte blanca tendrá un valor de  $1$ .

Nótese que la simplicidad de los filtros de Haar, hace que puedan calcularse convoluciones en tiempo constante haciendo uso de la imagen integrales calculada al comiendo de todo el algoritmo.

### Cálculo de la orientación

Una vez realizado el procedimiento descrito en el apartado anterior, conviene entender los pares obtenidos como puntos en el plano. Sobre dichos puntos se ejecuta un algoritmo que, informalmente, funciona de la siguiente manera.

1. Desde el origen, se trazan dos semirrectas que forman un ángulo  $\frac{\pi}{3}$ . Todos los puntos que se encuentren entre ambas semirrectas se suman (como vectores), almacenándose este vector.
2. Las rectas descritas en el anterior paso, van rotando, volviéndose a calcular la suma y almacenándose el resultado solamente si el vector obtenido es el de mayor norma hasta el momento.

La dirección del único vector almacenado por el algoritmo anterior es la orientación asignada al punto de interés. La figura 2.14 ilustra el procedimiento.

**Observación 2.2.1** (Poda de los extremos). Usualmente, solo se sumarán los puntos que, además de estar comprendidos entre las dos rectas, se encuentren dentro de un círculo de cierto radio predefinido, para evitar que los puntos con norma muy alta alteren en exceso el resultado.  $\diamond$

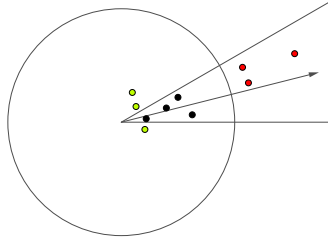


Figura 2.14: Ilustración del procedimiento de cálculo de la orientación. En rojo se muestran los puntos que, a pesar de encontrarse entre las dos semirrectas, no se tienen en cuenta para el cálculo del vector de orientación.

### U-SURF

El cálculo de la orientación de los puntos de interés es un procedimiento costoso, por tanto solo debe ser ejecutado cuando interese tener un descriptor invariante por rotaciones.

Si no es el caso, es muy recomendable ejecutar la versión de SURF que omite esta fase (U-SURF), y que, experimentalmente, se ha visto que tolera rotaciones de hasta  $\pm 15^\circ$ , tal y como se dice en [6].

### 2.2.2. Construcción del descriptor del punto de interés

Dado un punto de interés, lo primero que se hace es trazar un cuadrado de lado  $20s$  orientado en dirección a la orientación calculada para dicho punto. Tras esto, dicho cuadrado es dividido de manera que se forme un tablero de  $4 \times 4$  subcuadrados.

Sobre cada uno de los subcuadrados se escogen  $5 \times 5$  píxeles equiespaciados. De cada uno de estos píxeles se calcula, al igual que en la subsección anterior, sus derivadas parciales discretas ponderadas con una gaussiana centrada en el punto de interés. El lado de los filtros de Haar es  $2s + 1$ , mientras que la gaussiana tiene parámetro  $\sigma = 3,3s$ .

Una vez hechos estos cálculos, sobre cada subcuadrado se calculan los siguientes datos, siendo  $S$  el conjunto de  $5 \times 5$  píxeles escogido en el subcuadrado y  $\tilde{D}_x$  la derivada horizontal (análogamente vertical) ponderada con la gaussiana.

$$\begin{aligned}
 |D_x| &:= \sum_{(x,y) \in S} |\tilde{D}_x(x,y)| \\
 D_x &:= \sum_{(x,y) \in S} \tilde{D}_x(x,y) \\
 |D_y| &:= \sum_{(x,y) \in S} |\tilde{D}_y(x,y)| \\
 D_y &:= \sum_{(x,y) \in S} \tilde{D}_y(x,y)
 \end{aligned} \tag{2.21}$$

De esta forma, a cada subcuadrado le corresponden 4 números, y, al cuadrado completo, 64. Estos 64 datos se agrupan en forma de vector, y, este vector es el descriptor del punto de interés.

**Observación 2.2.2** (Distinción entre concavidad y convexidad). Además de calcular el determinante Hessiano de cada punto, SURF calcula también la traza de la matriz Hessiana, que, al ser invariante por cambios de referencia ortogonales, es la suma de los autovalores de la matriz Hessiana.

Por ende, si esta suma es positiva, la imagen es convexa en ese punto. Análogamente, si la traza es negativa, la imagen es cóncava en ese punto. Este cálculo adicional puede usarse para acelerar el proceso de búsqueda de correspondencias, ya que dos puntos correspondientes deben tener traza del mismo signo.  $\diamond$

**Observación 2.2.3** (Invarianza por rotaciones). Nótese que el descriptor de un punto de interés es invariante por rotaciones (siempre que no se haya ejecutado U-SURF), al estar el cuadrado sobre el que se calcula el descriptor rotado según la orientación del punto.  $\diamond$

**Variantes**

Se han propuesto diversas variantes en el cálculo del descriptor de SURF, algunas de ellas están basadas en hacer subdivisiones más gruesas o más finas del cuadrado inicial, obteniendo descriptores con más o menos componentes. La más utilizada de estas variantes es SURF-36, que divide el cuadrado en un tablero de tamaño  $3 \times 3$ .

Otras variantes se basan en calcular más datos en cada subcuadrado, por ejemplo, la variante SURF-128 se basa en calcular los datos de las ecuaciones (2.21) por duplicado, una vez para las derivadas positivas y otra para las derivadas negativas.

Algunos resultados experimentales fruto de la aplicación de estos métodos pueden consultarse en [6].



## Capítulo 3

# Emparejamiento de puntos

En el capítulo 2 se presentó el operador SURF, que detecta y describe puntos de interés sobre una imagen. A lo largo de este capítulo se asume que se dispone de dos imágenes, estando cada una de ellas dotada de un conjunto de puntos de interés junto con sus descriptores asociados.

El objetivo de este capítulo es presentar dos de las técnicas más usadas para encontrar un conjunto de pares de puntos de interés de ambas imágenes que se puedan considerar correspondientes.

### 3.1. Algoritmo de fuerza bruta

Una opción es, simplemente, comparar todos los descriptores  $\mathbf{x}$  de la primera imagen con todos los descriptores  $\mathbf{y}$  de la segunda, por ejemplo, usando la distancia euclídea como métrica.

Una vez hecho esto, las correspondencias vienen dadas por el conjunto ordenado  $\mathcal{M}$  de pares de descriptores ordenados ascendentemente en función de la distancia entre ellos. La única dificultad que entraña el algoritmo de fuerza bruta es que debe detectar y descartar pares de descriptores incompatibles. Nótese que un descriptor a lo sumo puede quedar asociado a otro, aunque puede ocurrir que no tenga correspondencia.

Existen diversas modificaciones aplicables al algoritmo de fuerza bruta, por ejemplo, quedarse únicamente con los  $K$  primeros pares de descriptores de  $\mathcal{M}$ . Otra opción es solo considerar como candidatos a emparejamientos válidos los pares de  $\mathcal{M}$  cuya distancia no supera cierto umbral  $T$ .

La figura 3.1 ilustra los resultados obtenidos con esta técnica.

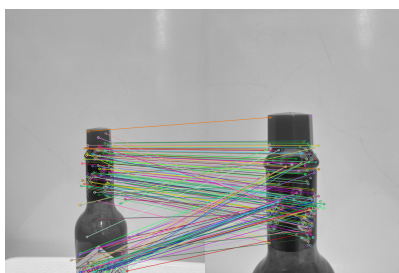


Figura 3.1: Ilustración de los emparejamientos encontrados por un algoritmo de fuerza bruta sin restricciones de umbral o selección de los  $K$  mejores. Nótese que, a pesar de que ambas imágenes únicamente difieren en un cambio de escala, hay multitud de emparejamientos incorrectos.

### 3.2. Regla del cociente de Lowe y comprobación cruzada

Puede darse el caso de que haya varios descriptores muy cercanos a uno dado, esto es, con propiedades similares. Esto puede deberse, por ejemplo, a la existencia de elementos repetidos en una imagen.

En tal caso, no está claro cuál es el par de descriptores correspondientes, por tanto, para evitar la aparición de falsas correspondencias, conviene descartar dicho descriptor.

El criterio de descarte propuesto por Lowe en [16], y cuyos resultados se muestran en la figura 3.2, consiste en, para cada descriptor  $\mathbf{x}$ , calcular sus dos descriptores más cercanos  $\mathbf{y}_1$ ,  $\mathbf{y}_2$ . Tras esto, si se verifica que  $\frac{\|\mathbf{x}-\mathbf{y}_1\|}{\|\mathbf{x}-\mathbf{y}_2\|} > \tau$ , se descarta el descriptor  $\mathbf{x}$ , en caso contrario, se considerará que el par de descriptores  $(\mathbf{x}, \mathbf{y}_1)$  es correspondiente. En [16] se propone escoger  $\tau = 0,8$ .

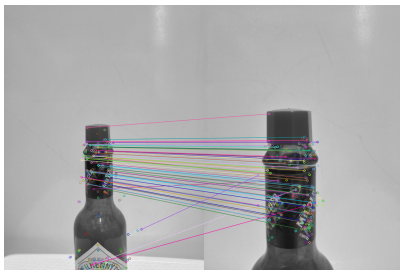


Figura 3.2: Ilustración de los emparejamientos que superan criba propuesta por Lowe. El número de emparejamientos incorrectos se reduce mucho.

Una alternativa a la regla del cociente de Lowe es la llamada **comprobación cruzada**, cuyos resultados ilustra la figura 3.3. Según esta regla, un par de descriptores  $(\mathbf{x}, \mathbf{y})$  solo serán considerados correspondientes si  $\mathbf{x}$  es el descriptor más cercano a  $\mathbf{y}$  y  $\mathbf{y}$  es el descriptor más cercano a  $\mathbf{x}$ .

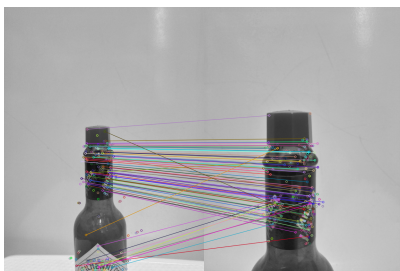


Figura 3.3: Ilustración de los emparejamientos que verifican el criterio de la comprobación cruzada. Los resultados son similares a los de la aplicación de la regla de Lowe.

### 3.3. Resultados sobre imágenes no adaptadas

Como se ha indicado previamente, las figuras 3.1, 3.2, 3.3 muestran resultados de los algoritmos de emparejamiento presentados para imágenes tomadas pensando en el comportamiento de SURF. Sin embargo, es interesante comprobar qué resultados se obtienen para un par de imágenes con menos restricciones, por ejemplo las mostradas en la figura 3.4.

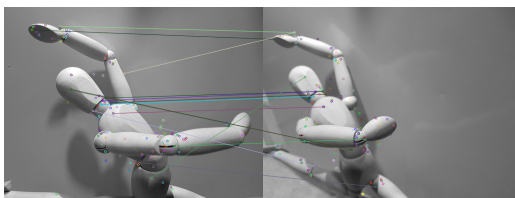


Figura 3.4: Ilustración de los emparejamientos que verifican el criterio de Lowe. Se observa que, ante cambios de inclinación, se obtienen escasos emparejamientos, siendo algunos de ellos erróneos.

## Capítulo 4

# Extensión afín

En el capítulo 2 se describe cómo el operador SURF, al igual que SIFT, simula cambios tanto en la escala como en la orientación de las imágenes que analiza. Este hecho garantiza correspondencias de puntos de interés robustas entre imágenes cuya principal diferencia pueda atribuirse a cambios de escala, rotaciones y traslaciones<sup>1</sup>.

No obstante, es usual el caso en el cual se quieren encontrar correspondencias entre imágenes que, además de diferir en términos de escala, orientación y traslación, difieren en la “inclinación” con la que han sido tomadas. En este capítulo se expone una extensión de los operadores SURF y SIFT que mejora la calidad de las correspondencias de puntos de interés en esta situación más general. Dicha extensión es conocida como ASIFT (**A**ffine-**SIFT**) [29], [26], [24].

### 4.1. Idea del algoritmo

Al contrario de lo que pasa con la escala y la orientación, no es posible simular de forma directa los cambios que se producirían en una imagen si ésta hubiera sido tomada con una cámara con un **eje principal** diferente. Esto se debe a que no se tiene información acerca de la profundidad de los puntos.

No obstante, si se asume que la profundidad es uniforme, es decir, que el objeto que muestra la imagen es plano, sí que es posible calcular de manera exacta la deformación producida por un cambio en el eje principal.

En concreto, usando conceptos que se analizarán en el capítulo 5, se puede demostrar que esta deformación es una homografía plana, la cual, asumiendo que el **centro óptico** de la cámara se encuentra lo suficientemente alejado, puede aproximarse por una afinidad.

**Teorema 4.1.1** (Descomposición). *Dada una afinidad entre planos con determinante positivo cuya matriz asociada es  $A$ , entonces, dicha matriz admite la siguiente factorización*

$$A = \lambda R_\psi \begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix} R_\phi \quad (4.1)$$

siendo  $\lambda > 0$  y  $t \in [0, 1]$  y  $R_\psi, R_\phi$  matrices de rotación.

*Demostración.* Es un caso particular de descomposición en valores singulares de una matriz. ■

Del teorema 4.1.1 se extrae que la distorsión producida por un cambio en el eje principal, que se aproxima mediante una afinidad, depende de un parámetro de escala  $\lambda$ , de dos ángulos  $\psi$  y  $\phi$  y de un factor de inclinación  $t$ .

Si la matriz de inclinación de la ecuación (4.1) y  $R_\phi$  conmutaran, el único parámetro a simular por el algoritmo sería  $t$ , ya que la afinidad quedaría totalmente determinada por una inclinación junto con una rotación y un cambio de escala, encargándose SURF de simular estos dos últimos.

---

<sup>1</sup>Este es el caso, por ejemplo, de la **visión estereoscópica**.

Como, en general, dichas matrices no conmutan, el algoritmo debe simular, tanto las rotaciones dadas por el ángulo  $\phi$  como las inclinaciones dadas por el factor  $t$ , encargándose SURF de simular las rotaciones y cambios de escala restantes.

Esto concuerda con la intuición geométrica acerca de lo que supone un cambio de eje principal. Esto se debe a que, una vez fijado un punto sobre el objeto plano, hay una biyección entre todos los posibles ejes principales y la semiesfera (sin su ecuador). Además, los puntos de la semiesfera, y, por extensión, los ejes principales, están determinados unívocamente por dos parámetros, las coordenadas polares de los puntos sobre la semiesfera (latitud y longitud). La figura 4.1 ilustra esta explicación.

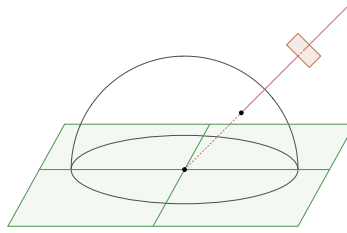


Figura 4.1: Ilustración del objeto plano (verde) junto a un nuevo eje principal (rojo) y su plano imagen asociado (marrón).

En resumen, lo que hace el algoritmo ASIFT es, sobre la imagen a procesar, aplicar transformaciones de la forma

$$\begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix} \mathbf{R}_\phi \mathbf{x} = \begin{pmatrix} t \cos \phi & -\sin \phi \\ t \sin \phi & \cos \phi \end{pmatrix} \mathbf{x} \quad (4.2)$$

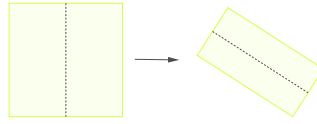


Figura 4.2: Ilustración del efecto de las transformaciones aplicadas por ASIFT. En la práctica, dichas transformaciones se realizan mediante técnicas de submuestreo.

La elección de los parámetros de las transformaciones a aplicar es libre, sin embargo, en [24] se recomienda escoger una sucesión de inclinaciones de la forma  $\{a^n\}_{n=0}^N$  con  $a < 1$ , y, para cada una de esas inclinaciones calcular las transformaciones para una sucesión de ángulos de la forma  $\{\frac{kb}{a^n}\}_{k=0}^K$ , siendo  $K$  el mayor entero que verifica que  $\frac{kb}{a^n} < \pi$ . En particular, en la implementación propuesta por [29] se toma  $N = 5$ ,  $a = \frac{1}{\sqrt{2}}$  y  $b = 0,4\pi$ .

Tras el cálculo de estas transformaciones, ASIFT aplicará un operador de detección y descripción de puntos de interés invariante por traslaciones, cambios de escala y rotaciones, por ejemplo, SURF.

Por último, a la hora de encontrar correspondencias de puntos de interés, ya no solo se realizarán comparaciones entre los puntos de interés de una y otra imagen, sino entre los puntos de interés de cada una de las transformaciones de la primera imagen con los puntos de interés de cada una de las transformaciones de la otra imagen. Esta es la fase más costosa en términos computacionales del algoritmo ASIFT.

## 4.2. Resultados

A continuación se muestran los resultados obtenidos mediante la aplicación del algoritmo ASIFT sobre algunos pares de imágenes de prueba. El código ejecutado para la generación de imágenes es el proporcionado por [29], al no estar disponible ASIFT en [2].



Las figuras 4.3 y 4.4 muestran una aplicación de ASIFT sobre imágenes de objetos planos, es decir, el ámbito donde las hipótesis que justifican el funcionamiento de ASIFT se cumplen. En concreto, las imágenes proporcionan diferentes perspectivas de una versión del cuadro *“Los tiempos fundidos de Isadora”*, cuyo original es obra de Eduardo Naranjo.

Las imágenes de la figura 4.3 tardaron 29 segundos en procesarse, habiéndose invertido 13 segundos en la detección y descripción de puntos de interés y 16 segundos en la búsqueda de correspondencias, siendo los resultados similares en el resto de las figuras utilizadas en los diferentes experimentos realizados.



Figura 4.3: Ilustración de las correspondencias entre una vista frontal y otra perspectiva.



Figura 4.4: Ilustración de las correspondencias entre dos perspectivas no frontales.

La suposición, en el marco teórico, de que la profundidad de los puntos sobre la imagen es constante puede parecer muy drástica, pues en la inmensa mayoría de los casos no será así, sin embargo, tal y como muestra la figura 4.5, en la práctica cosecha buenos resultados.

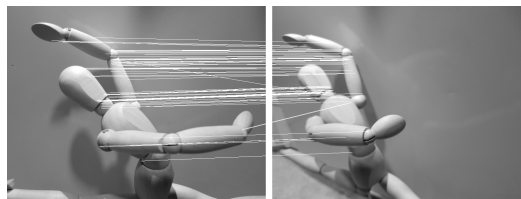


Figura 4.5: Ilustración de las correspondencias entre dos imágenes que muestran objetos no planos. Hay una clara mejora, tanto en calidad, como en cantidad, respecto de los emparejamientos encontrados en la figura 3.4.



# Parte II

## Reconstrucciones



## Capítulo 5

# La cámara proyectiva finita

Durante este capítulo, y en lo que resta de documento,  $\mathbb{P}^n$  representará el espacio proyectivo canónico complejo de dimensión  $n$ , es decir,  $\mathbb{P}(\mathbb{C}^{n+1})$ .

La función de una cámara es transformar una escena tridimensional en una imagen plana. El objetivo de este capítulo es presentar un modelo matemático de cámara, la **cámara proyectiva finita**.

Una cámara proyectiva finita es una transformación proyectiva  $\phi : \mathbb{P}^3 \rightarrow \mathbb{P}^2$  que cumple ciertas propiedades.  $\mathbb{P}^3$  se identifica con el conjunto de puntos del espacio, mientras que  $\mathbb{P}^2$  representa el conjunto de puntos de la imagen.

### 5.1. Modelo geométrico

El modelo geométrico de la cámara proyectiva finita es el siguiente. Dado un espacio proyectivo  $\mathbb{P}^3$ , se selecciona un punto  $c \in \mathbb{P}^3$  y un plano  $P \subset \mathbb{P}^3$  que no pasa por  $c$ . Hecho esto, se considera la aplicación

$$\begin{aligned} \phi : \mathbb{P}^3 \setminus \{c\} &\rightarrow P \\ x &\mapsto xc \cap P \end{aligned} \quad (5.1)$$

Como  $\phi$  es un caso particular de **proyección cónica**,  $\phi$  está bien definida, es decir,  $xc \cap P$  es siempre unipuntual, y,  $\phi$  es una aplicación proyectiva.

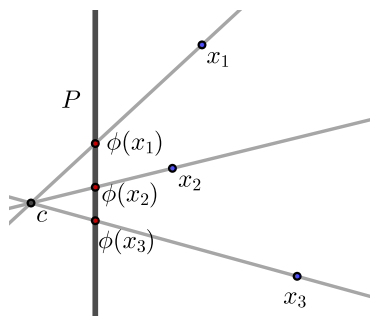


Figura 5.1: Ilustración bidimensional de la acción de  $\phi$  sobre algunos puntos del espacio.

Al punto  $c$  se le llama **centro** de  $\phi$ , mientras que el plano  $P$  usualmente recibe el nombre de **retina** o **plano imagen** de  $\phi$ .

#### 5.1.1. Referencias de la retina

Como  $\phi$  es una aplicación proyectiva, una vez fijadas ciertas referencias de  $\mathbb{P}^3$  y de  $P$ , esta queda totalmente determinada por una matriz  $3 \times 4$  única salvo múltiplos. En la sección 5.2 se calculan algunas matrices de cámaras. Para ello, se recurre a ciertas hipótesis sobre las referencias de  $P$ , las cuales se estudian en esta subsección.

En la práctica, la escena tridimensional que captura una cámara tiene estructura euclídea. Expresado en términos teóricos, el espacio proyectivo  $\mathbb{P}^3$  en el que se encuentra la escena, está dotado de estructura conforme. Es decir, han sido prefijados cierto plano  $\pi_\infty \neq P$  de manera que  $c \notin \pi_\infty$ <sup>1</sup> al que se llama **plano del infinito** y cierta cónica real imaginaria<sup>2</sup> no degenerada  $\Omega_\infty$  a la que se llama **cónica del absoluto**.

Esta estructura conforme del espacio ambiente induce una estructura conforme en la retina de la cámara. En efecto, la recta del infinito de  $P$  vendría dada por  $l_\infty := \pi_\infty \cap P$  y la cónica de absoluto por los dos puntos  $\{i, j\} := \Omega_\infty \cap P$ .

**Observación 5.1.1.** Es sencillo comprobar que, en efecto,  $\Omega_\infty \cap P$  consta de dos puntos. Esto se debe a que  $\Omega_\infty \cap P = \Omega_\infty \cap l_\infty$ , que se corresponde con el conjunto de soluciones de una ecuación polinómica real de segundo grado, la conocida como **ecuación de Joachimstahl**<sup>3</sup>, luego  $\Omega_\infty \cap l_\infty$  constará de 1 o 2 elementos.

No obstante, en el caso de que dicha ecuación tenga una solución doble, esta debe ser real, lo cual entra en contradicción con el hecho de que  $\Omega_\infty$  no posee puntos reales.  $\diamond$

A pesar de este hecho, en la práctica, la retina de las cámaras viene dotada de manera preestablecida (por la forma de sus píxeles) de una referencia proyectiva real  $\mathcal{R}_P$  que no tiene por qué ser conforme respecto a  $\{i, j\}$ . No obstante, dicha referencia siempre es afín respecto de  $l_\infty$ . A continuación se clasifican las familias de referencias notables que puede tener la retina.

### Referencias conformes

Las referencias más sencillas que puede tener una retina son aquellas que son conformes respecto de  $\{i, j\}$ . Es decir, referencias  $\mathcal{R}_P = \{x_0, x_1, c_P; e_P\}$  de manera que  $x_0, x_1 \in l_\infty$  y una matriz de  $\{i, j\}$  es la identidad  $2 \times 2$ . Dichas referencias pueden construirse geoméricamente de la siguiente manera.

1. Tomar un  $x_0 \in l_\infty$  arbitrario.
2. Seleccionar el cuarto armónico de  $x_0$  respecto de  $\{i, j\}$ . Por ejemplo, mediante la construcción de un cuadrilátero completo, tal y como se muestra en la figura 5.2.
3. Tomar un punto  $c_P \notin l_\infty$  arbitrario.
4. Para escoger un punto unidad adecuado, se puede seguir el procedimiento expuesto en el teorema B.2.1.

De las conclusiones extraídas del apéndice B se deduce que los píxeles de una cámara que posee una referencia conforme son cuadrados.

Un caso particular de estas referencias surge cuando, en lugar de tomar un punto  $c_P \notin l_\infty$  sin restricciones adicionales, se toma el punto de corte entre  $P$  y la recta perpendicular a  $P$  que pasa por el centro  $c$  de la cámara. A estas últimas referencias se las denomina **referencias canónicas de la retina**, mientras que al punto  $c_P$  de las mismas se le conoce como **punto principal** de  $\phi$  y a la recta  $cc_P$  como **eje principal** de  $\phi$ .

### Referencias afines con ejes ortogonales

Estas referencias son una generalización del caso anterior. El proceso de construcción de las mismas es idéntico, con la salvedad de que el punto unidad  $e_P$  se escoge arbitrariamente de entre los puntos que no están sobre las rectas que conforman el triángulo cuyos vértices son  $x_0, x_1$  y  $c_P$ .

A diferencia del caso anterior, las cámaras que poseen esta clase de referencias tienen los píxeles rectangulares.

<sup>1</sup>Los modelos de cámara en los que  $c \in \pi_\infty$  se denominan **cámaras afines**, pues son afinidades  $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ .

<sup>2</sup>Esto es, que tiene alguna matriz real pero ningún punto real.

<sup>3</sup> $(\mathbf{x} + \theta\mathbf{y})^\top \mathbf{A}(\mathbf{x} + \theta\mathbf{y}) = 0$ , es una ecuación de segundo grado en  $\theta$ .



## 5.2. Matrices de cámara

En esta sección se calculan las matrices de cámara para cada una de las familias de referencias expuestas.

En todos los casos se dispone de una referencia de la retina  $\mathcal{R}_P$  junto con su referencia adaptada  $\mathcal{R}$  de  $\mathbb{P}^3$ . Por tanto, se pueden considerar las bases asociadas de ambas referencias, que son únicas salvo múltiplos  $\mathcal{B} = \{u_1, u_2, u_3, u_4\}$  y  $\mathcal{B}_P = \{v_1, v_2, v_3\}$ .

Por tanto, la matriz de la cámara  $\phi$  será de la forma

$$\left( \begin{array}{c|c|c|c} \widehat{\phi}(u_1)_{\mathcal{B}_P} & \widehat{\phi}(u_2)_{\mathcal{B}_P} & \widehat{\phi}(u_3)_{\mathcal{B}_P} & \widehat{\phi}(u_4)_{\mathcal{B}_P} \end{array} \right) \quad (5.2)$$

siendo  $\widehat{\phi}$  la aplicación lineal asociada a  $\phi$  y  $\widehat{\phi}(u_i)_{\mathcal{B}_P}$  la matriz columna que representa las coordenadas de  $\widehat{\phi}(u_i)$  respecto de  $\mathcal{B}_P$ .

### Cálculo de la expresión analítica de $\phi$

En primer lugar, es conveniente calcular la expresión analítica de  $\phi$  respecto de la referencia adaptada de  $\mathbb{P}^3$ . Para ello, se toma un punto genérico  $x := (\alpha : \beta : \gamma : \delta)_{\mathcal{R}}$ , con lo que la recta que pasa por el centro óptico y por  $x$ , a la que se llamará  $xc$  viene dada por

$$\{(\alpha : \beta : \gamma : \delta + \theta)_{\mathcal{R}} : \theta \in \mathbb{C}\} \cup \{c\} \quad (5.3)$$

Luego, para que un punto de  $xc$  esté en  $P$ , se debe cumplir que  $\gamma = (\delta + \theta)f$ . Despejando  $\theta$  y sustituyendo se tiene que

$$\phi(x) = \left( \alpha : \beta : \gamma : \frac{\gamma}{f} \right)_{\mathcal{R}} \quad (5.4)$$

### Coordenadas del punto principal

A continuación, se procede a calcular las coordenadas del punto principal. Para ello, se toma un punto genérico  $x \in P \setminus \pi_{\infty}$  y se considera la recta  $xc$ .

Como  $x$  es un punto finito de  $P$ , sus coordenadas homogéneas normalizadas respecto de  $\mathcal{R}$  serán  $(\alpha : \beta : f : 1)_{\mathcal{R}}$ , luego una parametrización de  $xc$  es

$$\{(\alpha : \beta : f : 1 + \theta)_{\mathcal{R}} : \theta \in \mathbb{C}\} \cup \{c\} \quad (5.5)$$

La dirección de esta recta es el punto  $d := (\alpha : \beta : f : 0)_{\mathcal{R}}$ , y, para que  $xc$  sea perpendicular a  $P$ , dicho punto debe ser conjugado con todos los puntos de  $l_{\infty}$  respecto de  $\Omega_{\infty}$ . Para que esto se cumpla, basta con que  $d$  sea conjugado con  $x_0$  y  $x_1$ , ya que generan  $l_{\infty}$ . Esto último se cumple si y solo si  $\alpha = \beta = 0$ . Por ende, las coordenadas homogéneas del punto principal son  $(0 : 0 : f : 1)_{\mathcal{R}}$ .

### Matriz respecto de una referencia canónica

Como la referencia de  $\mathbb{P}^3$  y la referencia de  $P$  comparten los dos primeros puntos se cumple,

$$\begin{aligned} \widehat{\phi}(u_1) &= u_1 \sim v_1 \\ \widehat{\phi}(u_2) &= u_2 \sim v_2 \end{aligned} \quad (5.6)$$

Pero además, como ambas referencias son conformes, tanto en una como en otra, la matriz de la cónica del absoluto es la identidad, luego se puede suponer que  $u_1 = v_1$  y que  $u_2 = v_2$ . Además, por la ecuación (5.4) se sabe que

$$\widehat{\phi}(u_3) = u_3 + \frac{1}{f}u_4 \sim \frac{1}{f}v_3 \quad (5.7)$$

Siendo cierta la última proporcionalidad por ser  $v_3$  un representante del punto principal de  $\phi$ . Luego, como  $\widehat{\phi}(u_4) = 0$ , una matriz de  $\phi$  es proporcional a

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f}\lambda & 0 \end{pmatrix} \quad (5.8)$$



**Matriz respecto de una referencia conforme**

La única diferencia respecto al caso de la referencia canónica es que la ecuación (5.7) no se cumple, verificándose en cambio

$$\widehat{\phi}(u_3) = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 \quad (5.9)$$

con  $\alpha_3 \neq 0$ , pues  $y_2 \in \pi_\infty \setminus P$ , luego la matriz de  $\phi$  es proporcional a

$$\begin{pmatrix} 1 & 0 & \alpha_1 & 0 \\ 0 & 1 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 & 0 \end{pmatrix} \quad (5.10)$$

**Matriz respecto de una referencia afín con ejes ortogonales**

Como la referencia  $\mathcal{R}_P$  no es conforme, ya no se puede continuar suponiendo que  $u_1 = v_1$  y que  $u_2 = v_2$ , de modo que únicamente puede garantizarse que se verifican las ecuaciones (5.6), y, por tanto, la matriz de  $\phi$  es proporcional a

$$\begin{pmatrix} \lambda_1 & 0 & \alpha_1 & 0 \\ 0 & \lambda_2 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 & 0 \end{pmatrix} \quad (5.11)$$

**Matriz respecto de una referencia afín general**

En el caso de una referencia afín general las ecuaciones (5.6) no se cumplen, pudiéndose afirmar únicamente que

$$\begin{aligned} \widehat{\phi}(u_1) &= u_1 \sim v_1 \\ \widehat{\phi}(u_2) &= u_2 = \mu_1 v_1 + \mu_2 v_2 \end{aligned} \quad (5.12)$$

Siendo  $\mu_2 \neq 0$ , con lo que la matriz de  $\phi$  es proporcional a

$$\begin{pmatrix} \lambda & \mu_1 & \alpha_1 & 0 \\ 0 & \mu_2 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 & 0 \end{pmatrix} \quad (5.13)$$

Usando una notación más compacta, se define

$$K := \begin{pmatrix} \lambda & \mu_1 & \alpha_1 \\ 0 & \mu_2 & \alpha_2 \\ 0 & 0 & \alpha_3 \end{pmatrix} \quad (5.14)$$

que es una matriz triangular superior invertible (en su diagonal no hay ningún elemento nulo) a la que nos referiremos como **matriz intrínseca** de la cámara. Con esta definición, la matriz de una cámara  $\phi$  puede escribirse como  $K(I|0)$ .

**5.3. Matriz respecto de una referencia no adaptada**

Si se prescindiera de la referencia adaptada  $\mathcal{R}$  de  $\mathbb{P}^3$ , tomando otra referencia conforme cualquiera  $\mathcal{R}'$ , las coordenadas de los puntos respecto de una y otra referencia estarían relacionadas por una semejanza de matriz  $S$ . Dicho de otra manera, se verifica la igualdad

$$\mathbf{x}_{\mathcal{R}} \sim S \mathbf{x}_{\mathcal{R}'} \quad (5.15)$$

Por tanto, las ecuaciones que describen el comportamiento de  $\phi$  respecto de esta referencia no adaptada son

$$\mathbf{x}_{\mathcal{R}_P} \sim K(I|0) S \mathbf{x}_{\mathcal{R}'} \quad (5.16)$$

Como  $\mathbf{S}$  es la matriz de una semejanza, su matriz será de la forma

$$\left( \begin{array}{c|c} \alpha \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0}^\top & 1 \end{array} \right) \quad (5.17)$$

siendo  $\mathbf{R}$  una matriz ortogonal  $3 \times 3$ . Es claro que si  $(\tilde{\mathbf{c}}|1)^\top$  son las coordenadas homogéneas del centro respecto de  $\mathcal{R}'$ , entonces  $(\mathbf{0}^\top|1)^\top = \mathbf{S}(\tilde{\mathbf{c}}|1)^\top$ . Desarrollando esta ecuación es posible calcular  $\mathbf{t}$ . En efecto

$$\left( \begin{array}{c|c} \alpha \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0}^\top & 1 \end{array} \right) \left( \begin{array}{c} \tilde{\mathbf{c}} \\ 1 \end{array} \right) = \left( \begin{array}{c} \mathbf{0} \\ 1 \end{array} \right) \iff \left( \begin{array}{c} \alpha \mathbf{R} \tilde{\mathbf{c}} + \mathbf{t} \\ 1 \end{array} \right) = \left( \begin{array}{c} \mathbf{0} \\ 1 \end{array} \right) \iff \mathbf{t} = -\alpha \mathbf{R} \tilde{\mathbf{c}} \quad (5.18)$$

Luego, retomando la ecuación (5.16) se obtiene que

$$\mathbf{x}_{\mathcal{R}_P} \sim \mathbf{K}(\mathbf{I}|\mathbf{0}) \left( \begin{array}{c|c} \alpha \mathbf{R} & -\alpha \mathbf{R} \tilde{\mathbf{c}} \\ \hline \mathbf{0}^\top & 1 \end{array} \right) \mathbf{x}_{\mathcal{R}'} \sim \mathbf{K} \mathbf{R}(\mathbf{I} | -\tilde{\mathbf{c}}) \mathbf{x}_{\mathcal{R}'} \quad (5.19)$$

A la matriz  $\mathbf{R}$  y al vector  $\tilde{\mathbf{c}}$  se les suele llamar **parámetros extrínsecos** de la cámara  $\phi$ , pues no dependen de la aplicación en sí, sino de la referencia escogida del espacio.

**Teorema 5.3.1.** *Toda matriz  $\mathbf{P} = (\mathbf{M}|\mathbf{m})$  de dimensión  $3 \times 4$  con  $\mathbf{M}$  invertible es la matriz de cierta cámara proyectiva finita.*

*Demostración.* En efecto,  $\mathbf{P} = \mathbf{M}(\mathbf{I}|\mathbf{M}^{-1}\mathbf{m})$ , y, como  $\mathbf{M}$  es una matriz cuadrada  $3 \times 3$  se puede factorizar como producto de una matriz triangular superior  $\mathbf{K}$  y una matriz ortogonal  $\mathbf{R}$ . Esta factorización puede realizarse mediante el algoritmo descrito en la sección A.1<sup>4</sup>.

De esta manera  $\mathbf{P} = \mathbf{K} \mathbf{R}(\mathbf{I} | -\tilde{\mathbf{c}})$ , siendo  $-\tilde{\mathbf{c}} = \mathbf{M}^{-1}\mathbf{m}$ . ■

**Observación 5.3.1.** Nótese que si  $\mathbf{P}$  es una matriz de cámara y  $\mathbf{c}$  es un vector proporcional a las coordenadas del centro de la misma,  $\mathbf{P}\mathbf{c} = \mathbf{0}$ . En efecto

$$\mathbf{P}\mathbf{c} = \mathbf{K} \mathbf{R}(\mathbf{I} | -\tilde{\mathbf{c}}) \left( \begin{array}{c} \tilde{\mathbf{c}} \\ 1 \end{array} \right) = \mathbf{K} \mathbf{R} \mathbf{0} = \mathbf{0} \quad (5.20)$$

Luego el centro de la cámara también es el centro de la aplicación proyectiva que la modela. ◇

---

<sup>4</sup>Se conseguiría el mismo efecto aplicando el método de Gram-Schmidt, no obstante, numéricamente es más inestable.

## Capítulo 6

# Geometría epipolar

Cuando se proyecta una escena tridimensional sobre un plano mediante una cámara proyectiva, se está perdiendo mucha información, en concreto, se pierde la profundidad de los puntos proyectados, es decir, la tercera dimensión de la escena.

Recuperar esta información a partir de una única imagen no es posible geoméricamente, sin embargo, sí es posible realizar reconstrucciones de una escena tridimensional a partir de dos imágenes de la misma. Para ello, es necesario conocer las relaciones geométricas existentes entre las dos cámaras que captan la escena.

A lo largo de este capítulo no se asume que las cámaras tengan relación alguna entre sí, no obstante, en el capítulo 9 de [12] pueden encontrarse estudios detallados sobre situaciones particulares de interés (traslación pura, movimientos en un plano,...).

### 6.1. Introducción

Sean dos cámaras proyectivas finitas  $\phi_1, \phi_2$  con centros distintos y un punto  $x_1 \in P_1$  tal que  $x_1 \notin c_1c_2$ . Es claro que  $x_1 = \phi_1(x)$ , siendo  $x \neq c_1$  cualquier punto sobre la recta  $x_1c_1$ . Dicho esto, es conveniente preguntarse qué se puede decir de los puntos  $x_2 \in P_2$  que verifiquen que  $x_2 = \phi_2(x)$  con  $x \in x_1c_1$ , es decir, qué se puede decir de la proyección mediante  $\phi_2$  de los puntos de la imagen inversa de  $x_1$ . La figura 6.1 ofrece una ilustración del problema.

Como, por hipótesis,  $x_1 \notin c_1c_2$ , los puntos  $c_1, c_2$  y  $x_1$  engendran un plano  $\pi$ , al que se llama **plano epipolar** de  $x_1$ . Para generalizar esta noción se introduce la siguiente definición.

**Definición 6.1.1** (Haz de planos epipolares). Se llama **haz de planos epipolares** al conjunto de planos de  $\mathbb{P}^3$  que pasan por la recta que une los centros de  $\phi_1$  y  $\phi_2$ , es decir, la recta  $c_1c_2$ , a la que se llama **recta base**. Es importante notar que el haz de planos epipolares puede ser visto como una recta en el espacio dual  $\mathbb{P}^{3*}$ .

En la figura 6.2 se ilustran un par de planos  $\pi, \pi'$  del haz de planos epipolares.

**Proposición 6.1.1.** La imagen obtenida mediante  $\phi_2$  de  $x_1c_1$  es  $\pi \cap P_2$ , siendo  $\pi$  el plano epipolar.

*Demostración.* Es claro que  $x_1c_1$ , está contenida en  $\pi$ . Si  $x$  es un punto arbitrario de  $x_1c_1$ , por definición,  $\phi_2(x) = c_2x \cap P_2$ . Luego, como  $c_2x \subset \pi$ , se tiene que  $\phi_2(x) \in \pi \cap P_2$ .

Recíprocamente, dado un punto  $x_2 \in \pi \cap P_2$ , su imagen inversa es  $x_2c_2 \setminus \{c_2\} \subset \pi$ , que, por tanto, se corta con  $x_1c_1$  en cierto punto  $x$ . De esta manera,  $x_2 = \phi_2(x)$ , tal y como se quería demostrar. ■

**Definición 6.1.2** (Recta epipolar). Dado un punto  $x_1 \in P_1$ , se llama **recta epipolar** asociada a  $x_1$  a la recta  $\pi \cap P_2$ , siendo  $\pi$  el plano epipolar asociado a  $x_1$ . Por la fórmula de Grassman<sup>1</sup> y las hipótesis consideradas, está garantizado que la recta epipolar es, efectivamente, una recta.

<sup>1</sup> $\dim(\langle A, B \rangle) = \dim(A) + \dim(B) - \dim(A \cap B)$

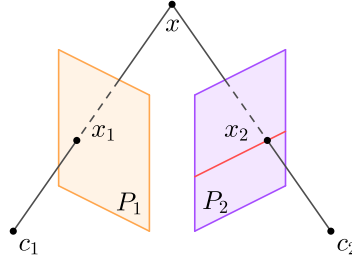


Figura 6.1: Ilustración (en rojo) de la recta epipolar de  $x_1$ .

**Definición 6.1.3** (Epipolos). Se definen los **epipolos** como los puntos de intersección de la recta base con los planos imagen  $P_1$  y  $P_2$ . Nótese que el conjunto de las rectas epipolares de un plano imagen constituyen un haz de rectas con punto base en el epipolo correspondiente. Esto se debe a que cualquier plano epipolar contiene a la recta base.

En un contexto más realista, un punto en el plano imagen de una cámara está asociado a un único punto del espacio, el primer punto opaco que se encuentra.

Si se tiene en mente esta consideración adicional, el planteamiento de esta sección se puede resumir diciendo que, dado un punto  $x_1$  de un plano imagen, el punto del espacio  $x$  que éste representa se proyecta sobre el segundo plano imagen en algún punto de la recta epipolar de  $x_1$ .

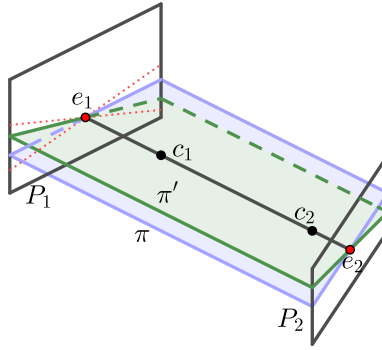


Figura 6.2: Ilustración (en rojo) de los epipolos  $e_1$  y  $e_2$  como base de los haces de rectas epipolares.

## 6.2. Obtención de la recta epipolar

En esta sección se trata de hallar una expresión explícita para la función que a cada punto de un plano imagen le asigna su recta epipolar en el otro plano imagen, así como determinar su naturaleza.

Esta aplicación puede ser vista como una función  $P_1 \cong \mathbb{P}^2 \rightarrow P_2^* \cong \mathbb{P}^{2*}$ . Esto se debe a que las rectas de un plano pueden ser vistas como puntos en su dual.

A continuación se expone un método geométrico con el que construir la aplicación requerida. Éste consta de dos pasos, en primer lugar, dado un punto  $x_1 \in P_1$ , se hallará un punto  $x_2$  de su recta epipolar. Tras esto, se obtendrá el epipolo  $e_2 \in P_2$ , siendo la recta epipolar simplemente  $e_2x_2$ .

El primer paso del método se realiza mediante una técnica de “*transferencia por un plano*”, de la siguiente manera

1. Se considera un plano  $\pi$  que no pase por ninguno de los centros de cámara.
2. La imagen inversa de  $x_1$  es una recta que pasa por  $c_1$  de la cual se ha suprimido  $c_1$ . Por tanto, dicha recta se corta con  $\pi$  en un solo punto, al que se llamará  $\tilde{x}$ .

3. Dicho punto  $\tilde{x}$  se proyectará mediante  $\phi_2$  a cierto punto  $x_2$  de la recta epipolar de  $x_1$ .

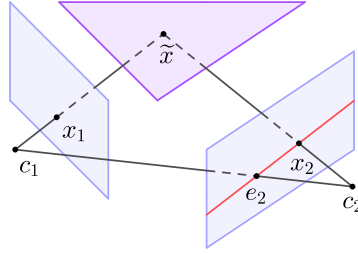


Figura 6.3: Ilustración del método de transferencia por un plano.

La siguiente proposición muestra que la aplicación (dependiente del plano  $\pi$  escogido) que describe el proceso anterior es una homografía.

**Proposición 6.2.1.** *La aplicación definida en la ecuación (6.1) es una homografía*

$$\begin{aligned} P_1 &\rightarrow P_2 \\ x_1 &\mapsto \phi_2(\phi_1^{-1}(x_1) \cap \pi) \end{aligned} \quad (6.1)$$

*Demostración.* En efecto, la aplicación descrita en (6.1), se puede ver como la composición de la proyección cónica punto-plano con centro  $c_1$  y plano  $\pi$  (restringida a  $P_1$ ) con la proyección cónica  $\phi_2$  restringida a  $\pi$ . Con esto tenemos que es una aplicación proyectiva.

Como  $c_1 \notin P_1$ , la primera proyección aplicada es inyectiva. Análogamente, como  $c_2 \notin \pi$ , entonces  $\phi_2|_{\pi}$  es inyectiva. Luego la composición de ambas proyecciones es, a su vez, inyectiva. ■

Como conclusión de esta última proposición se obtiene que, fijadas ciertas referencias de  $P_1$  y  $P_2$ , la aplicación definida en (6.1) queda determinada por una matriz regular  $3 \times 3$  única salvo múltiplos. Se denotará a dicha matriz por  $H_{\pi}$ .

Por ende, el último paso a seguir sería hallar el epipolo, simplemente  $\phi_2(c_1)$ , y determinar el punto de  $P_2^*$  que se corresponde con la recta  $e_2x_2$ . Para esto último es necesario recordar el siguiente resultado.

**Lema 6.2.2.** *Fijada una referencia de  $\mathbb{P}^2$ , dados dos puntos  $x_1 \neq x_2$ , las coordenadas del punto dual correspondiente con la recta  $x_1x_2$  (respecto de la referencia dual) son  $\mathbf{l} \sim \mathbf{x}_1 \times \mathbf{x}_2$ . Además se verifica que  $x_1x_2$  tiene por ecuación implícita  $\mathbf{l}^T \mathbf{x} = 0$ .*

Luego, una vez fijada una referencia de  $P_2$ , el punto de  $P_2^*$  correspondiente con la recta epipolar, escrito en términos de la referencia dual, está representado por el vector  $\mathbf{e}_2 \times \mathbf{x}_2$ .

**Lema 6.2.3.** *La aplicación que, a cada punto  $x$  de un plano le asocia la recta formada por dicho punto y otro punto  $\tilde{x}$  previamente fijado es una aplicación proyectiva.*

*Demostración.* Por el lema 6.2.2 se sabe que, una vez fijada una referencia, la recta  $\tilde{x}x$  viene dada por  $\tilde{\mathbf{x}} \times \mathbf{x}$ . Si  $[\tilde{\mathbf{x}}] \sim (\alpha : \beta : \gamma)$ , este producto puede expresarse matricialmente como

$$[\tilde{\mathbf{x}}]_{\times} \mathbf{x} \sim \begin{pmatrix} 0 & -\gamma & \beta \\ \gamma & 0 & -\alpha \\ -\beta & \alpha & 0 \end{pmatrix} \mathbf{x} \quad (6.2)$$

Luego se tiene el resultado, siendo la matriz asociada, salvo múltiplos,  $[\tilde{\mathbf{x}}]_{\times}$ . ■

Así pues, una vez fijada una referencia de  $P_1$ , se tiene que la recta epipolar de cierto punto  $x_1 \in P_1$  viene dada por coordenadas proporcionales a

$$\mathbf{F}\mathbf{x}_1 = ([\mathbf{e}_2]_{\times} H_{\pi})\mathbf{x}_1 \quad (6.3)$$

Luego la aplicación buscada es una aplicación proyectiva, y, además, es una aplicación proyectiva de rango 2, por ser su espacio de partida un plano y su imagen una recta en el dual (haz de rectas epipolares de  $P_2$  con base  $e_2$ ).

El inconveniente de la expresión obtenida en la ecuación (6.3) es que es dependiente de cierto plano auxiliar  $\pi$ , cuya única utilidad era la de ayudar a encontrar un punto en la imagen inversa de  $x_1$ . La siguiente proposición proporciona una manera alternativa de encontrar dicho punto.

**Proposición 6.2.4.** *Si, fijadas ciertas referencias,  $\phi_1$  tiene por matriz a  $P_1$ , la imagen inversa de un punto  $x_1 \in P_1$  viene dada por las ecuaciones paramétricas*

$$\{ [P_1^+ x_1 + \theta c_1] : \theta \in \mathbb{C} \} \quad (6.4)$$

donde  $P_1^+ := P_1^\top (P_1 P_1^\top)^{-1}$  es la matriz pseudoinversa de  $P_1$ . Nótese que  $P_1 P_1^+ = I$ .

*Demostración.* En efecto, la ecuación (6.4) se corresponde con las ecuaciones paramétricas de una recta que pasa por  $c_1$ , sin incluir a este punto.

Además, proyectando cualquier punto de la recta se obtiene que

$$[P_1(P_1^+ x_1 + \theta c_1)] = [P_1 P_1^+ x_1 + \theta P_1 c_1] = [x_1 + \theta 0] = [x_1] \quad (6.5)$$

como se quería demostrar. ■

Luego, una vez fijadas ciertas referencias, la recta epipolar asociada al punto  $x_1$  vendrá dada por los puntos  $P_2(P_1^+ x_1)$  y el epipolo  $P_2 c_1$ . Es decir, viene dada (salvo múltiplos) por la expresión

$$F x_1 := ([P_2 c_1] \times P_2 P_1^+) x_1 \quad (6.6)$$

Diremos que la matriz  $F$  definida en (6.6) es la **matriz asociada** al par de cámaras  $(\phi_1, \phi_2)$ .

**Definición 6.2.1** (Puntos correspondientes). Dados  $x_1 \in P_1$  y  $x_2 \in P_2$ . Se dirá que son **correspondientes** si hay un punto  $x \in \mathbb{P}^3$  que cumpla que  $\phi_1(x) = x_1$  y  $\phi_2(x) = x_2$ .

**Proposición 6.2.5.** *Dados dos puntos correspondientes  $x_1, x_2$ . Entonces, la matriz  $F$  asociada a ambas cámaras verifica la siguiente igualdad*

$$x_2^\top F x_1 = 0 \quad (6.7)$$

*Demostración.* Si  $x_1 \in P_1$  y  $x_2 \in P_2$  se identifican con el mismo punto  $x \in \mathbb{P}^3$ , es claro que  $x_2$  está en la recta epipolar de  $x_1$ , que tiene por ecuación  $(F x_1)^\top y = 0$ , luego se verifica que  $(F x_1)^\top x_2 = 0$ . Si se traspone esta última igualdad queda que  $x_2^\top F x_1 = 0$ . ■

**Corolario 6.2.6.** *Dados dos puntos  $x_1 \in P_1$  y  $x_2 \in P_2$  tales que verifican la ecuación (6.7), se cumple que las imágenes inversas  $x_1$  y  $x_2$  respecto de las cámaras correspondientes son coplanarias.*

*Demostración.* Si se cumple que  $x_2^\top F x_1 = 0$  entonces  $x_2$  está en la recta epipolar de  $x_1$ , luego es la proyección de cierto punto de la imagen inversa de  $x_1$ . Por ende, ambas imágenes inversas (rectas salvo un punto) se cortan, luego son coplanarias. ■

### 6.3. Matriz fundamental

**Definición 6.3.1** (Matriz fundamental). Dadas dos cámaras  $\phi_1$  y  $\phi_2$ , se dirá que  $F$  es una **matriz fundamental** asociada al par  $(\phi_1, \phi_2)$  si, para cada par de puntos correspondientes  $(x_1, x_2) \in P_1 \times P_2$  se verifica

$$x_2^\top F x_1 = 0 \quad (6.8)$$

**Lema 6.3.1.** *Si  $F$  es matriz fundamental de  $(\phi_1, \phi_2)$ , entonces  $F^\top$  lo es de  $(\phi_2, \phi_1)$ .*

*Demostración.* Dados dos puntos correspondientes  $(x_1, x_2)$ , se cumple que  $x_2^\top F x_1 = 0$  luego basta trasponer la igualdad numérica para obtener que  $x_1^\top F^\top x_2 = 0$ , como se quería. ■

Es evidente por la proposición 6.2.5 que la matriz asociada a un par de cámaras definida en (6.6) es una matriz fundamental de dicho par de cámaras.

**Proposición 6.3.2.** *Sea  $F$  una matriz fundamental asociada a  $(\phi_1, \phi_2)$ . Dado un punto  $x_1 \in P_1$ , se cumple que  $(F\mathbf{x}_1)^\top \mathbf{y} = 0$  es la ecuación de la recta epipolar asociada a  $x_1$ .*

*Demostración.* En efecto, veamos que, al proyectar cualquier punto  $x$  de la imagen inversa de  $x_1$  mediante  $\phi_2$  se obtiene un punto que verifica la ecuación  $(F\mathbf{x}_1)^\top P_2\mathbf{x} = 0$ .

$$(F\mathbf{x}_1)^\top P_2\mathbf{x} = \mathbf{x}_1^\top F^\top (P_2\mathbf{x}) \quad (6.9)$$

Como  $x_1$  y  $\phi_2(x)$ , son, por definición, correspondientes y  $F^\top$  es una matriz asociada del par  $(\phi_2, \phi_1)$ , se verifica que  $\mathbf{x}_1^\top F^\top (P_2\mathbf{x}) = 0$ , y, por tanto, se tiene el resultado. ■

Como la matriz asociada a un par de cámaras es la única matriz (salvo múltiplos) que verifica que a cada punto le asocia su recta epipolar, se tiene que las nociones de matriz fundamental y matriz asociada son equivalentes.

**Observación 6.3.1** (Núcleo de  $F$ ). Anteriormente se dedujo que  $F$  tiene rango 2, sin embargo, no se encontró explícitamente ningún punto de su núcleo. No obstante, es inmediato comprobar que  $\mathbf{e}_1$  está en el núcleo de  $F$ . En efecto

$$F\mathbf{e}_1 = [\mathbf{e}_2]_\times P_2(P_1^+ \mathbf{e}_1) \quad (6.10)$$

Siendo  $P_1^+ \mathbf{e}_1$  un punto de la recta  $c_1c_2$ , luego  $P_2(P_1^+ \mathbf{e}_1) = P_2(\mathbf{c}_1) = \mathbf{e}_2$ . Por último,  $[\mathbf{e}_2]_\times \mathbf{e}_2 = \mathbf{e}_2 \times \mathbf{e}_2 = \mathbf{0}$ . ◇

## 6.4. Homografía epipolar

Hasta ahora se ha construido una aplicación tal que, a cada punto de  $P_1 \setminus \{e_1\}$  le asigna su recta epipolar. Un hecho importante, es que si dos puntos  $x_1, y_1$  son colineales con  $e_1$ , sus rectas epipolares coinciden.

En efecto, por ser  $y_1 \neq e_1$  un punto de la recta  $x_1e_1$ , se tiene que  $\mathbf{y}_1 \sim \mathbf{x}_1 + \theta\mathbf{e}_1$ , con  $\theta \in \mathbb{C}$ . Por lo tanto

$$F\mathbf{y}_1 \sim F(\mathbf{x}_1 + \theta\mathbf{e}_1) = F\mathbf{x}_1 + \theta F\mathbf{e}_1 = F\mathbf{x}_1 \quad (6.11)$$

Este resultado indica que hay una correspondencia entre los haces de rectas epipolares de  $P_1$  y  $P_2$ , es decir, una aplicación

$$l_1 \subset P_1^\star \cong \mathbb{P}^{2\star} \rightarrow l_2 \subset P_2^\star \cong \mathbb{P}^{2\star} \quad (6.12)$$

siendo  $l_1$  y  $l_2$  los haces de rectas epipolares de  $P_1$  y  $P_2$  respectivamente, que pueden ser vistos como rectas en los espacios duales correspondientes.

Para obtener una expresión explícita de dicha aplicación, se aplicará la función dada por la matriz fundamental (definición 6.3.1) a puntos específicos de cada recta del haz. En particular, es claro que si se toma una recta que no pase por  $e_1$ , dicha recta cortará a todas las rectas del haz en un único punto distinto de  $e_1$ . Una elección estándar de esta recta puede ser la recta  $\mathbf{e}_1^\top \mathbf{x} = 0$ , la cual, trivialmente, no pasa por  $e_1$ .

Para obtener las intersecciones entre esta recta y las rectas del haz es necesario recordar el siguiente lema.

**Lema 6.4.1.** *Si  $\mathbf{a}^\top \mathbf{x} = 0$  y  $\mathbf{b}^\top \mathbf{x} = 0$  son dos rectas de  $\mathbb{P}^2$ , entonces, el punto de intersección de las mismas tiene coordenadas proporcionales a  $\mathbf{a} \times \mathbf{b}$ .*

Por tanto, si  $\mathbf{r}_1^\top \mathbf{x} = 0$  es una recta del haz, entonces  $\mathbf{p} := [\mathbf{e}_1 \times \mathbf{r}_1]$  es un punto de dicha recta distinto de  $e_1$ . Luego, si la recta epipolar asociada a  $\mathbf{p}$  viene dada por  $\mathbf{r}_2^\top \mathbf{x} = 0$  se tiene que

$$\mathbf{r}_2 \sim (F[\mathbf{e}_1]_\times) \mathbf{r}_1 \quad (6.13)$$

De la ecuación (6.13) se deduce que la correspondencia buscada es una aplicación proyectiva de rango 1 entre dos rectas, por tanto, una homografía. La figura 6.4 ilustra esta correspondencia.





*Demostración.* Por la ecuación (6.6) se sabe que  $F = [P_2 c_1]_{\times} P_2 P_1^+$ . El único obstáculo es conocer  $c_1$ , sin embargo, se sabe que  $P_1 c_1 = 0$ , por ser el centro de  $\phi_1$ , además, al estar trabajando con cámaras finitas,  $c_1 = (\widetilde{c}_1^T | 1)^T$ . Por tanto

$$P_1 c_1 = (I|0)(\widetilde{c}_1^T | 1)^T = \widetilde{c}_1 = 0 \implies c_1 = (0^T | 1)^T \quad (6.18)$$

Sustituyendo en la definición de  $F$  y multiplicando por bloques se obtiene el resultado. ■

En esta subsección se ha concluido que un par de cámaras tiene una única matriz fundamental, y, sin embargo, puede haber varios pares de cámaras con la misma matriz fundamental, tal y como se demuestra en el corolario 6.5.3.

**Lema 6.5.2.** *Dados los pares de cámaras  $(\phi_1, \phi_2)$ ,  $(\phi_3, \phi_4)$  tales que, fijadas ciertas referencias, vienen representadas por las matrices  $P_1, P_2, P_1 H$  y  $P_2 H$  respectivamente, siendo  $H$  una matriz  $4 \times 4$  invertible. Existe una biyección entre los pares de puntos correspondientes de  $(\phi_1, \phi_2)$  y los de  $(\phi_3, \phi_4)$ .*

*Demostración.* Si  $(x_1, x_2) \in P_1 \times P_2$  son puntos correspondientes de  $(\phi_1, \phi_2)$ , hay un punto  $x \in \mathbb{P}^3$  tal que  $\phi_1(x) = x_1$  y  $\phi_2(x) = x_2$ . Escribiendo en términos de las referencias correspondientes

$$P_1 x = x_1 \text{ y } P_2 x = x_2 \quad (6.19)$$

Los puntos  $(x_3, x_4) \in P_3 \times P_4$  cuyas coordenadas respecto de las referencias escogidas de  $P_3$  y  $P_4$  son proporcionales a  $x_1$  y  $x_2$ , respectivamente, son puntos correspondientes de  $(\phi_3, \phi_4)$ . En efecto, habría que probar que hay un punto  $x'$  de manera que  $\phi_3(x') = x_3$  y  $\phi_4(x') = x_4$ . Escrito en coordenadas, se debe cumplir que  $P_1 H x' = x_1$  y que  $P_2 H x' = x_2$ . Luego basta con tomar como  $x'$  el punto de  $\mathbb{P}^3$  cuyas coordenadas son proporcionales a  $H^{-1}x$ .

De esta manera, la biyección entre pares de puntos correspondientes es la identidad. ■

**Corolario 6.5.3.** *Dados dos pares de cámaras en las condiciones del lema 6.5.2, se cumple que su matriz fundamental es la misma.*

*Demostración.* Si  $F$  es la matriz fundamental de  $(\phi_1, \phi_2)$ , por el lema 6.5.2 se tiene que si los puntos con coordenadas proporcionales a  $x_3$  y  $x_4$  son puntos correspondientes de  $(\phi_3, \phi_4)$ , los puntos de  $P_1$  y  $P_2$  con coordenadas proporcionales a  $x_3$  y  $x_4$  son puntos correspondientes de  $(\phi_1, \phi_2)$ , luego  $x_4^T F x_3 = 0$ , por tanto, por definición,  $F$  es la matriz fundamental de  $(\phi_3, \phi_4)$ . ■

El corolario 6.5.3 proporciona una forma de producir pares de cámaras distintos con la misma matriz fundamental. De hecho, todos los pares de cámaras con dicha propiedad se pueden generar de esa forma, tal y como demuestra el corolario 6.5.6.

**Lema 6.5.4.** *Si una matriz fundamental  $F$  verifica la ecuación (6.22) se cumple que  $m' = \lambda m$  y que  $M' = \lambda^{-1}(M + m v^T)$  para ciertos  $\lambda$  y  $v$ .*

*Demostración.* Se da la igualdad

$$m^T F = m^T [m]_{\times} M = (-[m]_{\times} m)^T M = (m \times m)^T M = 0^T M = 0^T \quad (6.20)$$

Por tanto, análogamente,  $m'^T F = 0^T$ . Y, como  $F$  es una matriz de rango 2,  $m \sim m'$ , luego  $m' = \lambda m$  para cierto  $\lambda$ . Por otra parte, como se cumple la ecuación (6.22) se tiene que

$$\lambda [m]_{\times} M' - [m]_{\times} M = 0 \iff [m]_{\times} (\lambda M' - M) = 0 \quad (6.21)$$

Como las columnas del último producto de la ecuación son los productos vectoriales de  $m$  por cada una de las columnas de  $\lambda M' - M$ , y todos son nulos, se concluye que todas las columnas de  $\lambda M' - M$  deben ser proporcionales a  $m$ . O sea que  $\lambda M' - M = m v^T$  para cierto  $v$ . ■

**Teorema 6.5.5.** *Dados dos pares de cámaras que comparten matriz fundamental, y que, además, una vez fijadas ciertas referencias, vienen dadas por las matrices  $((I|0), (M|m))$  y  $((I|0), (M'|m'))$ .*

*Hay una matriz regular  $H$  tal que  $(I|0)H \sim (I|0)$  y  $(M|m)H = (M'|m')$ .*

*Demostración.* Si  $\mathbf{F}$  es la matriz fundamental asociada a los pares de cámaras que vienen dadas por las matrices  $((\mathbf{I}|\mathbf{0}), (\mathbf{M}|\mathbf{m}))$  y  $((\mathbf{I}|\mathbf{0}), (\mathbf{M}'|\mathbf{m}'))$ , se tiene, por el lema 6.5.1, la siguiente igualdad

$$\mathbf{F} = [\mathbf{m}]_{\times} \mathbf{M} = [\mathbf{m}']_{\times} \mathbf{M}' \quad (6.22)$$

Usando el lema 6.5.4 se tiene que  $(\mathbf{M}'|\mathbf{m}') = (\lambda^{-1}(\mathbf{M} + \mathbf{m}\mathbf{v}^{\top})|\lambda\mathbf{m})$ . Usando esto, es fácil ver que hay un matriz regular  $\mathbf{H}$  de modo que  $(\mathbf{M}|\mathbf{m})\mathbf{H} = (\mathbf{M}'|\mathbf{m}')$ . Basta con hallar una solución a la ecuación

$$(\mathbf{M}|\mathbf{m}')\mathbf{H} = (\mathbf{M}|\mathbf{m}') \left( \frac{\mathbf{A}}{\mathbf{c}^{\top}} \middle| \frac{\mathbf{b}}{d} \right) = (\mathbf{M}\mathbf{A} + \mathbf{m}\mathbf{c}^{\top}|\mathbf{M}\mathbf{b} + \mathbf{m}d) = (\lambda^{-1}(\mathbf{M} + \mathbf{m}\mathbf{v}^{\top})|\lambda\mathbf{m}) \quad (6.23)$$

Para lo cual basta con elegir  $\mathbf{A} = \lambda^{-1}\mathbf{I}$ ,  $\mathbf{b} = \mathbf{0}$ ,  $\mathbf{c}^{\top} = \lambda^{-1}\mathbf{v}^{\top}$  y  $d = \lambda$ .

Además,  $(\mathbf{I}|\mathbf{0})\mathbf{H} = \lambda^{-1}(\mathbf{I}|\mathbf{0}) \sim (\mathbf{I}|\mathbf{0})$ , con lo que se tiene el resultado. ■

**Corolario 6.5.6.** Sean dos pares de cámaras dados por las matrices  $(\mathbf{P}_1, \mathbf{P}_2)$  y  $(\mathbf{P}_3, \mathbf{P}_4)$  con la misma matriz fundamental, entonces hay una matriz regular  $\mathbf{Q}$  de manera que  $\mathbf{P}_1\mathbf{Q} = \mathbf{P}_3$  y  $\mathbf{P}_2\mathbf{Q} = \mathbf{P}_4$ .

*Demostración.* Es claro que  $\mathbf{P}_1\mathbf{G}\mathbf{H}\mathbf{G}'^{-1} \sim \mathbf{P}_3$  y que  $\mathbf{P}_2\mathbf{G}\mathbf{H}\mathbf{G}'^{-1} = \mathbf{P}_4$ . Siendo  $\mathbf{G}$  la matriz regular que hace que  $\mathbf{P}_1\mathbf{G} = (\mathbf{I}|\mathbf{0})$  y  $\mathbf{G}'$  la matriz regular que hace que  $\mathbf{P}_1\mathbf{G} = (\mathbf{I}|\mathbf{0})$ .

Como por el corolario 6.5.3, y por hipótesis, se tiene que los pares  $(\mathbf{P}_1, \mathbf{P}_2)$ ,  $(\mathbf{P}_1\mathbf{G}, \mathbf{P}_2\mathbf{G})$ ,  $(\mathbf{P}_3, \mathbf{P}_4)$  y  $(\mathbf{P}_3\mathbf{G}', \mathbf{P}_4\mathbf{G}')$  tienen todos la misma matriz fundamental, y, además, se puede aplicar el teorema 6.5.5 a los pares  $(\mathbf{P}_1\mathbf{G}, \mathbf{P}_2\mathbf{G})$  y  $(\mathbf{P}_3\mathbf{G}', \mathbf{P}_4\mathbf{G}')$ , se tiene que  $\mathbf{P}_1\mathbf{G}\mathbf{H} \sim \mathbf{G}'\mathbf{P}_3$  y que  $\mathbf{P}_2\mathbf{G}\mathbf{H} = \mathbf{G}'\mathbf{P}_4$ .

Despejando se obtiene el resultado buscado. ■

## 6.6. Matrices de cámara y matriz fundamental

Antes de comenzar se recuerda el siguiente resultado elemental.

**Lema 6.6.1.** Una matriz  $\mathbf{A}$  es antisimétrica si y solo si para todo  $\mathbf{x}$  se cumple que  $\mathbf{x}^{\top}\mathbf{A}\mathbf{x} = 0$ .

*Demostración.* Dado  $\mathbf{x}$ , se cumple que  $\mathbf{x}^{\top}\mathbf{A}\mathbf{x} = \mathbf{x}^{\top}\mathbf{A}^{\top}\mathbf{x}$ . Teniendo esto en cuenta, si  $\mathbf{A}$  es antisimétrica es claro que  $\mathbf{x}^{\top}\mathbf{A}\mathbf{x} = 0$ .

Recíprocamente, si  $\mathbf{x}^{\top}\mathbf{A}\mathbf{x} = 0$  para todo  $\mathbf{x}$ , se tiene que  $0 = (\mathbf{x} + \mathbf{y})^{\top}\mathbf{A}(\mathbf{x} + \mathbf{y}) = \mathbf{x}^{\top}(\mathbf{A} + \mathbf{A}^{\top})\mathbf{y}$ , luego  $\mathbf{A} + \mathbf{A}^{\top}$  representa a la forma bilineal idénticamente nula, y, por tanto, es la matriz nula, quedando que  $\mathbf{A}^{\top} = -\mathbf{A}$ . ■

**Proposición 6.6.2.** Una matriz no nula  $\mathbf{F}$  es matriz fundamental del par de cámaras dado por  $(\mathbf{P}_1, \mathbf{P}_2)$  si y solo si la matriz  $\mathbf{P}_2^{\top}\mathbf{F}\mathbf{P}_1$  es antisimétrica.

*Demostración.*  $\mathbf{P}_1^{\top}\mathbf{F}\mathbf{P}_2$  es antisimétrica si y solo si  $\mathbf{x}^{\top}\mathbf{P}_1^{\top}\mathbf{F}\mathbf{P}_2\mathbf{x} = 0$  para todo  $\mathbf{x}$ . Asimismo,  $\mathbf{F}$  es matriz fundamental si y solo si para cada par de puntos correspondientes  $x_1, x_2$  se cumple que  $\mathbf{x}_2^{\top}\mathbf{F}\mathbf{x}_1 = 0$ , y,  $(x_1, x_2)$  son correspondientes si y solo si hay un  $\mathbf{x}$  que cumple que  $\mathbf{x}_i = \mathbf{P}_i\mathbf{x}$ .

Así pues,  $\mathbf{F}$  es matriz fundamental si y solo si  $\mathbf{x}^{\top}\mathbf{P}_1^{\top}\mathbf{F}\mathbf{P}_2\mathbf{x} = 0$  para todo  $\mathbf{x}$ , como se quería. ■

Una consecuencia inmediata del corolario 6.5.6 es que, si el par de cámaras dado por las matrices  $((\mathbf{I}|\mathbf{0}), (\mathbf{M}|\mathbf{m}))$  tiene por matriz fundamental a  $\mathbf{F}$ , entonces, el conjunto de pares de cámaras con las que comparte matriz fundamental viene dado por

$$(\mathbf{I}|\mathbf{0})\mathbf{H} \quad (\mathbf{M} + \mathbf{m}\mathbf{v}^{\top}|\lambda\mathbf{m})\mathbf{H} \quad \text{para todo } \mathbf{v} \text{ y todo } \lambda > 0 \quad (6.24)$$

Usando la proposición 6.6.2 es sencillo comprobar que, dada una matriz fundamental  $\mathbf{F}$ , el par de cámaras dado por las matrices  $(\mathbf{I}|\mathbf{0})$  y  $([\mathbf{e}_2]_{\times}\mathbf{F}|\mathbf{e}_2)$  tiene por matriz fundamental a  $\mathbf{F}$ , y, por tanto, la familia de pares de cámaras que tienen a  $\mathbf{F}$  por matriz fundamental es

$$(\mathbf{I}|\mathbf{0})\mathbf{H} \quad ([\mathbf{e}_2]_{\times}\mathbf{F} + \mathbf{e}_2\mathbf{v}^{\top}|\lambda\mathbf{e}_2)\mathbf{H} \quad \text{para todo } \mathbf{v} \neq \mathbf{0} \text{ y todo } \lambda > 0 \quad (6.25)$$

**Observación 6.6.1.** Nótese que  $[\mathbf{e}_2]_{\times}\mathbf{F}$  tiene rango 2. Esto se debe a que sus columnas generan el plano ortogonal a  $\mathbf{e}_2$ . ◇

En efecto, se comprueba que  $\mathbf{A} := ([\mathbf{e}_2]_{\times} \mathbf{F} | \mathbf{e}_2)^{\top} \mathbf{F} (\mathbf{I} | \mathbf{0})$  es antisimétrica.

$$\begin{aligned} \mathbf{A}^{\top} &= \left( \frac{\mathbf{I}}{\mathbf{0}^{\top}} \right) \mathbf{F}^{\top} ([\mathbf{e}_2]_{\times} \mathbf{F} | \mathbf{e}_2) = \left( \frac{\mathbf{F}^{\top}}{\mathbf{0}^{\top}} \right) ([\mathbf{e}_2]_{\times} \mathbf{F} | \mathbf{e}_2) = \left( \frac{\mathbf{F}^{\top} [\mathbf{e}_2]_{\times} \mathbf{F} | \mathbf{0}}{\mathbf{0}^{\top} | 0} \right) \\ \mathbf{A} &= \left( \frac{\mathbf{F}^{\top} [\mathbf{e}_2]_{\times} \mathbf{F} | \mathbf{0}}{\mathbf{0}^{\top} | 0} \right)^{\top} = - \left( \frac{\mathbf{F}^{\top} [\mathbf{e}_2]_{\times} \mathbf{F} | \mathbf{0}}{\mathbf{0}^{\top} | 0} \right) = -\mathbf{A}^{\top} \end{aligned} \quad (6.26)$$

## 6.7. Recapitulación

Los resultados fundamentales de este capítulo son los siguientes.

1. Si se conocen las matrices de las cámaras  $\phi_1$  y  $\phi_2$ , es decir, si las cámaras están **calibradas**, se conoce la matriz fundamental, por la ecuación (6.6).
2. Si se desconocen las matrices de las cámaras pero se conoce la matriz fundamental  $\mathbf{F}$ , se pueden conocer una familia de pares de matrices de cámaras que comparten dicha matriz fundamental.

La situación más común es la 2, y es de la que se parte en el capítulo 7, donde se desarrollan técnicas para calcular la matriz fundamental a partir de pares de puntos correspondientes, que, en la práctica, vienen dados fruto de la ejecución de algoritmos como los descritos en la parte I de este documento.



## Capítulo 7

# Reconstrucciones

Dado un conjunto de pares de puntos correspondientes representados en coordenadas de  $P_1$  y  $P_2$  por  $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ , se quieren encontrar matrices de cámara  $P_1, P_2$  y puntos  $\{\mathbf{z}_n\}_{n=1}^N$  de manera que se cumpla que  $P_1 \mathbf{z}_n = \mathbf{x}_n$  y  $P_2 \mathbf{z}_n = \mathbf{y}_n$  para todo  $n \in \{1, \dots, N\}$ . A la terna  $(P_1, P_2, \{\mathbf{z}_n\}_{n=1}^N)$  se la denomina **reconstrucción** del conjunto de correspondencias  $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ .

### 7.1. Cómputo de la matriz fundamental

Dadas las cámaras  $\phi_1, \phi_2$  y un conjunto de pares de puntos correspondientes entre las mismas, es posible calcular la matriz fundamental de  $\phi_1$  y  $\phi_2$ , tal y como se expone a continuación.

Se puede asumir que, dados dos puntos correspondientes  $x_1 \in P_1$  y  $x_2 \in P_2$ , ambos son finitos, es decir, las coordenadas de  $x_i$  respecto de la referencia interna de  $P_i$  (desconocida) pueden normalizarse de manera que  $\mathbf{x}_i = (x_i \ y_i \ 1)$ .

Además, como  $x_1$  y  $x_2$  son correspondientes, se sabe por 6.3.1 que se verifica la igualdad  $\mathbf{x}_2^\top \mathbf{F} \mathbf{x}_1 = 0$ . Desarrollando esto se tiene

$$\mathbf{x}_2^\top \mathbf{F} \mathbf{x}_1 = 0 \iff \begin{pmatrix} x_2 & y_2 & 1 \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = 0 \iff \quad (7.1)$$
$$f_{11}x_1x_2 + f_{12}y_1x_2 + f_{13}x_2 + f_{21}x_1y_2 + f_{22}y_1y_2 + f_{23}y_2 + f_{31}x_1 + f_{32}y_1 + f_{33} = 0$$

En definitiva, cada par de puntos correspondientes induce una ecuación lineal sobre los coeficientes de  $\mathbf{F}$ . De esta forma, con  $n$  pares de correspondencias se obtiene un sistema de  $n$  ecuaciones y 9 incógnitas, luego, si se tienen al menos 8 pares de puntos correspondientes, la matriz fundamental puede ser determinada (salvo múltiplos) mediante cualquier algoritmo de resolución de sistemas lineales.

Sin embargo, como ya se vio en la figura 3.1, será relativamente frecuente encontrar pares de puntos que no conforman una correspondencia real. Por ende, es recomendable que el algoritmo usado para el cálculo de la matriz fundamental sea robusto ante este tipo de fallos.

Un método que cumple este requisito es el algoritmo RANSAC (**R**andom **S**ample **C**onsensus) [9] [12], el cual se expone a continuación.

**Definición 7.1.1** (Soporte). Se define el **soporte** de un conjunto de 8 correspondencias como el conjunto de pares de puntos correspondientes  $(x_1, x_2)$  que verifican que  $\mathbf{x}_2^\top \mathbf{F} \mathbf{x}_1 < \varepsilon^1$  para cierto umbral  $\varepsilon$ , siendo  $\mathbf{F}$  la matriz fundamental asociada a las 8 correspondencias seleccionadas.

La idea del algoritmo RANSAC consiste en tomar 8 correspondencias aleatorias y calcular, tanto la matriz fundamental asociada a éstas (mediante un algoritmo de resolución de sistemas lineales) como su soporte. Y, tras esto, y, a no ser que el cardinal del soporte supere cierto umbral, repetir el proceso un número de veces dado<sup>2</sup>. El resultado del algoritmo será la matriz fundamental con mayor

<sup>1</sup>En el capítulo 12 de [12] se proponen condiciones alternativas para que una correspondencia esté en el soporte.

<sup>2</sup>O elegido dinámicamente tal y como se explica en la sección 4.7 de [12].

soporte de entre las calculadas, considerándose, a partir de este momento solo las correspondencias que forman parte de su soporte.

**Observación 7.1.1.** Para refinar un poco más el resultado obtenido por RANSAC, se puede estimar una nueva matriz fundamental a partir de las correspondencias que pertenecen al soporte de la dada por RANSAC. Para realizar esta segunda estimación se suelen usar métodos no lineales como el algoritmo de Levenberg–Marquardt [12], [25].  $\diamond$

### Modificación para garantizar la singularidad de $F$

Una cuestión a tener en cuenta es que la matriz fundamental, tal y como se vio en 6.3.1, tiene rango 2. En general, el procedimiento descrito para calcular la matriz fundamental no garantiza que la matriz resultante tenga rango 2.

Una modificación sobre el algoritmo previamente expuesto que garantiza que la matriz resultante tiene rango 2 consiste, simplemente, en configurar RANSAC para que, en lugar de tomar 8 correspondencias de puntos, tome 7. De esta forma, usualmente, el conjunto de soluciones (salvo múltiplos) del sistema a resolver tendrá dimensión 1, quedando parametrizado por

$$\{\lambda F_1 + F_2\} \cup \{F_1\} \text{ con } \lambda \in \mathbb{R} \quad (7.2)$$

Se buscan por tanto, las matrices del conjunto dado por (7.2) con determinante nulo. Para hallar estas matrices es necesario primero hallar los  $\lambda \in \mathbb{R}$  que verifican la condición

$$\det(\lambda F_1 + F_2) = 0 \quad (7.3)$$

Nótese que la ecuación (7.3) es un polinomio de grado 3, y, por tanto, tendrá 1 o 3 soluciones reales. Por ende, por cada conjunto de correspondencias elegido por RANSAC, se deberá resolver un sistema de ecuaciones, una ecuación polinómica de tercer grado y calcular el soporte de hasta 3 matrices.

### Resultados

Para finalizar esta sección, es interesante notar que, una vez calculada la matriz fundamental, dado un punto de interés sobre  $P_1$ , se puede calcular su recta epipolar sobre  $P_2$ , tal y como se demostró en la ecuación (6.6). Realizar este ejercicio con dos imágenes reales puede resultar interesante, pues, entre otras cosas, permite visualizar gráficamente el epipolo de ambos planos imagen.

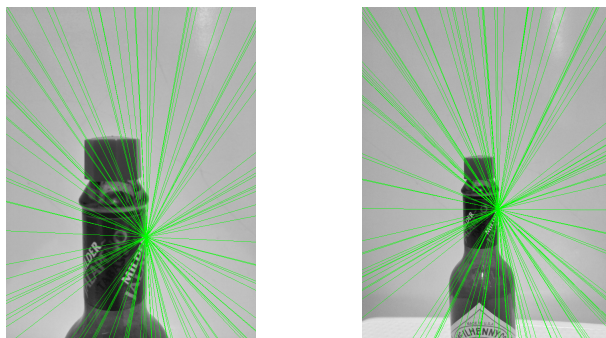


Figura 7.1: Ilustración de las rectas epipolares de ambos planos imagen. Dichas rectas son las correspondientes a las correspondencias sobre el soporte de la matriz fundamental calculada con RANSAC.

Nótese que la posición de los epipolos, que son los puntos de corte de los haces de rectas en la figura 7.1, coincide con la interpretación geométrica de los mismos, ya que, al diferir las dos imágenes mostradas únicamente en la escala, cabe esperar que los planos imagen de ambas cámaras sean casi paralelos, siendo por tanto la recta que une los centros ópticos aproximadamente perpendicular a ambos, quedando, por tanto, los epipolos situados, aproximadamente, en la misma posición en ambas imágenes.

## 7.2. Teorema de reconstrucción proyectiva

A continuación se presenta uno de los resultados clave de este documento.

**Teorema 7.2.1** (Reconstrucción proyectiva). *Dado un conjunto de al menos 8 correspondencias  $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$  y dos reconstrucciones  $(P_1, P_2, \{\mathbf{z}_n\}_{n=1}^N)$  y  $(P'_1, P'_2, \{\mathbf{z}'_n\}_{n=1}^N)$ , existe una matriz regular  $H$  de manera que  $P'_i = P_i H^{-1}$  para  $i = 1, 2$  y  $\mathbf{z}'_n = H \mathbf{z}_n$  para todo  $n$  salvo que  $\mathbf{x}_n$  e  $\mathbf{y}_n$  coincidan con los epipolos.*

*Demostración.* Tal y como se deduce en la sección 7.1, el conjunto de correspondencias  $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$  determina unívocamente la matriz fundamental  $F$  del par de cámaras al cual están asociadas.

Como los pares de matrices  $(P_1, P_2)$  y  $(P'_1, P'_2)$  comparten matriz fundamental, entonces, por el corolario 6.5.6, hay una matriz regular  $H$  de manera que  $P'_i = P_i H^{-1}$ .

Por otra parte, se verifican las ecuaciones

$$\begin{aligned} P'_1 H \mathbf{z}_n &= P_1 H^{-1} H \mathbf{z}_n = P_1 \mathbf{z}_n = \mathbf{x}_n \\ P'_1 \mathbf{z}'_n &= \mathbf{x}_n \end{aligned} \quad (7.4)$$

Por ende,  $H \mathbf{z}_n$  y  $\mathbf{z}'_n$  están sobre una misma recta que pasa por el centro de  $P'_1$ . Además, unas ecuaciones similares a (7.4) se deducen para  $P'_2$ . Por tanto,  $H \mathbf{z}_n$  y  $\mathbf{z}'_n$  están sobre una recta que pasa por el centro de  $P'_1$  y sobre una recta que pasa por el centro de  $P'_2$ . Por tanto, si  $\mathbf{x}_n$  e  $\mathbf{y}_n$  no coinciden con los epipolos, se deduce que  $\mathbf{z}'_n = H \mathbf{z}_n$ . ■

## 7.3. Cómputo de la triangulación

Dado un par de puntos correspondientes  $x_1 \in P_1$  y  $x_2 \in P_2$ , se quieren encontrar las coordenadas homogéneas  $\mathbf{x}$  de un punto  $x \in \mathbb{P}^3$  de manera que  $P_i \mathbf{x} \sim \mathbf{x}_i$ .

Si las correspondencias de puntos fueran perfectas, el cálculo de las coordenadas de  $\mathbf{x}$  podría realizarse simplemente mediante un procedimiento geométrico, tomando la intersección de las imágenes inversas de  $x_1$  y  $x_2$  vía  $\phi_1$  y  $\phi_2$ .

No obstante, si se trata de hacer esto en la práctica, usualmente ocurre que las imágenes inversas no se cortan. Para ver cómo resolver este problema, es necesario realizar el planteamiento algebraico del mismo.

Se sabe que  $x_i$  es un punto finito de  $P_i$ , luego sus coordenadas homogéneas pueden ser normalizadas a  $(x_i \ y_i \ 1)^\top$ . Además, se están buscando unas coordenadas  $\mathbf{x}$  de manera que  $P_i \mathbf{x} \sim \mathbf{x}_i$ . Equivalentemente,  $P_i \mathbf{x} \times \mathbf{x}_i = \mathbf{0}$ . Escribiendo

$$P_i \mathbf{x} = \begin{pmatrix} \mathbf{p}_{1i}^\top \mathbf{x} \\ \mathbf{p}_{2i}^\top \mathbf{x} \\ \mathbf{p}_{3i}^\top \mathbf{x} \end{pmatrix}, \quad (7.5)$$

donde  $\mathbf{p}_{ij}^\top$  representa a la fila  $i$ -ésima de la matriz  $P_j$ , y desarrollando el producto vectorial, se obtiene que el hecho de que unas coordenadas  $\mathbf{x}$  cumplan la restricción requerida es equivalente a que sean solución del siguiente sistema de ecuaciones lineal.

$$\begin{aligned} y_i \mathbf{p}_{3i}^\top \mathbf{x} - \mathbf{p}_{2i}^\top \mathbf{x} &= 0 \\ \mathbf{p}_{1i}^\top \mathbf{x} - x_i \mathbf{p}_{3i}^\top \mathbf{x} &= 0 \\ x_i \mathbf{p}_{3i}^\top \mathbf{x} - y_i \mathbf{p}_{1i}^\top \mathbf{x} &= 0 \end{aligned} \quad (7.6)$$

Nótese que la tercera ecuación de (7.6) es redundante, pues es combinación lineal de las dos primeras. Reuniendo las ecuaciones inducidas por las restricciones de ambas cámaras se obtiene el siguiente sistema

$$\mathbf{A} \mathbf{x} = \begin{pmatrix} y_1 \mathbf{p}_{31}^\top - \mathbf{p}_{21}^\top \\ \mathbf{p}_{11}^\top - x_1 \mathbf{p}_{31}^\top \\ y_2 \mathbf{p}_{32}^\top - \mathbf{p}_{22}^\top \\ \mathbf{p}_{12}^\top - x_2 \mathbf{p}_{32}^\top \end{pmatrix} \mathbf{x} = \mathbf{0} \quad (7.7)$$

Como normalmente las imágenes inversas de  $x_1$  y  $x_2$  no se cortan, el sistema (7.7) normalmente tendrá rango 4, y, por tanto, su única solución será la trivial. Por lo tanto, no es conveniente aplicar

un algoritmo para buscar soluciones de (7.7), sino encontrar un método para hallar un  $\mathbf{x}$  unitario que minimice  $\|\mathbf{Ax}\|^2$ , es decir, hallar una solución de mínimos cuadrados. De esta forma se podrán encontrar unas coordenadas representantes del punto proyectivo de  $\mathbb{P}^3$  que más se aproxima a ser una solución del sistema (7.7).

Para resolver este problema de minimización, se procede a calcular una descomposición en valores singulares (A.2) de  $\mathbf{A}$ , de manera que se tiene  $\mathbf{A} = \mathbf{UDV}^\top$ , luego se busca un  $\mathbf{x}$  unitario que haga mínima la norma  $\|\mathbf{UDV}^\top \mathbf{x}\|^2$ . Como  $\mathbf{U}$  es una matriz ortogonal, se cumple que  $\|\mathbf{UDV}^\top \mathbf{x}\|^2 = \|\mathbf{DV}^\top \mathbf{x}\|^2$ .

Asimismo, como  $\mathbf{V}$  es ortogonal y  $\|\mathbf{x}\| = 1$ , definiendo  $\mathbf{y} := \mathbf{V}^\top \mathbf{x}$ , se tiene que  $\|\mathbf{y}\| = 1$ . Por tanto, si se encuentra un  $\mathbf{y}$  sobre la esfera unidad que minimice  $\|\mathbf{Dy}\|^2$ , el problema de minimización original quedaría automáticamente resuelto.

El problema de minimizar  $\|\mathbf{Dy}\|^2$  sobre la esfera unidad es un problema de extremos condicionados fácilmente resoluble aplicando el teorema de los multiplicadores de Lagrange, es decir, minimizando la función lagrangiana

$$L(\mathbf{y}, \lambda) = (\mathbf{Dy})^\top \mathbf{Dy} + \lambda(\mathbf{yy}^\top - 1) \quad (7.8)$$

Para minimizar (7.8) es necesario calcular su gradiente respecto de  $\mathbf{y}$ , para lo cual se considera la función auxiliar

$$h_{\mathbf{a}}(\varepsilon) = (\mathbf{D}(\mathbf{y} + \varepsilon \mathbf{a}))^\top \mathbf{D}(\mathbf{y} + \varepsilon \mathbf{a}) + \lambda((\mathbf{y} + \varepsilon \mathbf{a})^\top (\mathbf{y} + \varepsilon \mathbf{a}) - 1) \quad (7.9)$$

A partir de la cual se calcula la derivada direccional de  $L(\mathbf{y}, \lambda)$  sobre la dirección  $\mathbf{a}$ .

$$h'_{\mathbf{a}}(0) = (2\mathbf{y}^\top \mathbf{D}^2 + 2\lambda \mathbf{y}^\top) \mathbf{a} \quad (7.10)$$

Luego se verifica que

$$\nabla_{\mathbf{y}} L(\mathbf{y}, \lambda) = 2\mathbf{y}^\top \mathbf{D}^2 + 2\lambda \mathbf{y}^\top \quad (7.11)$$

Por tanto, despejando, se obtiene que los puntos críticos de la función lagrangiana respecto de  $\mathbf{y}$  cumplen que

$$\mathbf{D}^2 \mathbf{y} = -\lambda \mathbf{y} \quad (7.12)$$

Es decir, una solución  $\mathbf{y}$  al problema de minimización debe ser autovector de  $\mathbf{D}^2$ , en concreto, para que sea un mínimo absoluto, debe ser el autovector asociado al menor autovalor  $-\lambda$  de  $\mathbf{D}^2$ .

Como  $\mathbf{D}$  es una matriz diagonal,  $\mathbf{D}^2$  también lo es, siendo, por tanto, los elementos (todos positivos) de su diagonal sus autovalores, que están ordenados de mayor a menor, por estarlo así también los elementos de la diagonal de  $\mathbf{D}$ .

Luego  $\mathbf{y}$  debe ser el autovector unitario asociado al autovalor correspondiente al último elemento de la diagonal de  $\mathbf{D}^2$ , es decir,  $\mathbf{y} = (0 \ 0 \ 0 \ 1)^\top$ . Además, como  $\mathbf{x} = \mathbf{Vy}$ ,  $\mathbf{x}$  resulta ser la última columna de  $\mathbf{V}$ .

## 7.4. Recapitulación

Llegados a este punto, ya se dispone de un algoritmo para el cálculo de una reconstrucción proyectiva a partir de un conjunto de correspondencias de puntos de interés. El algoritmo consta de los siguientes pasos

1. Calcula la matriz fundamental  $\mathbf{F}$  usando la técnica propuesta en la sección 7.1.
2. Calcula un par de matrices de cámara asociadas a  $\mathbf{F}$ , por ejemplo, usando la ecuación (6.25).
3. Triangula los puntos de la escena usando la técnica propuesta en 7.3.

La figura 7.2 representa a la nube de puntos tridimensionales (puntos blancos) obtenida tras la aplicación del algoritmo descrito en esta sección. La imagen en sí es una captura de la salida producida por el archivo “[reco.py](#)”.



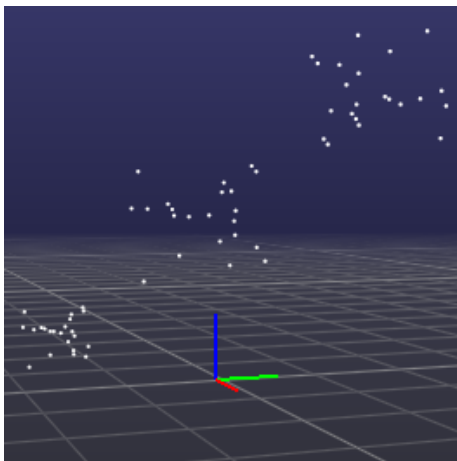


Figura 7.2: Reconstrucción proyectiva del par de imágenes 7.1, donde se ha usado un umbral Hessianiano de 500 para la detección de puntos de interés y la regla de Lowe para obtener emparejamientos de descriptores.



## Parte III

# Resultados y conclusiones



## Capítulo 8

# Resultados y conclusiones

Este capítulo consta de una recopilación de resultados, donde se comentan todos los detalles de los experimentos realizados, y una presentación de conclusiones, donde se evalúa la consecución de los objetivos planteados en la sección 1.2 y se plantean propuestas de trabajo futuro.

### 8.1. Resultados

Todos los algoritmos descritos en el documento han sido ejecutados sobre un dispositivo con las siguientes especificaciones

- Memoria RAM: 11,7 GiB.
- Procesador: Intel® Core™ i5-4210U CPU 1,70 GHz×4.
- Gráficos: Intel® Haswell Mobile.

Las fotografías han sido tomadas (sin *flash*) con una cámara Sony IMX 214 de 13 MPx, que se corresponde con la cámara trasera de un teléfono BQ Aquaris X5. Antes de ejecutar los algoritmos descritos en el documento sobre las imágenes tomadas, estas han sido re-escaladas (manteniendo la proporción) a resoluciones similares a  $800 \times 600$  píxeles.

Siempre se ha procurado que las imágenes tomadas no contuvieran elementos distintos a los objetos sobre los que se querían realizar experimentos.

#### El operador SURF

Tal y como se aprecia en las figuras 2.12 y 2.8, el detector SURF funciona según lo previsto, sin haber detectado anomalías como sí aparecían en algunos casos en [22], debido a la aparición de falsos bordes verticales y horizontales en las imágenes convolucionadas, lo cual es, a su vez, producto de la aproximación de los núcleos gaussianos con filtros caja.

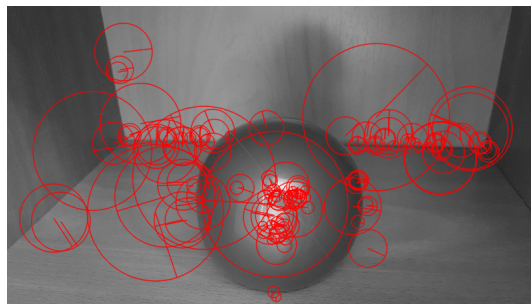


Figura 8.1: Ilustración de los puntos de interés encontrados por SURF. El tamaño de las circunferencias representa la escala a la que se detectó el punto, mientras que el radio dibujado de las mismas representa la orientación asignada.

En la figura 8.1 se aprecia el mismo efecto que en las figuras 2.12 y 2.8, es decir, una acumulación de puntos de interés sobre bordes y esquinas, así como sobre *blobs*.

Para poder hacer uso de la implementación algoritmo SURF disponible en [2] ha sido necesario compilar su código fuente con un *flag*<sup>1</sup> específico. Esta dificultad se debe a que SURF (al igual que SIFT) es un algoritmo patentado<sup>2</sup>, no siendo su uso libre para fines comerciales.

### Emparejamiento de puntos de interés

En las figuras 3.1, 3.3 y 3.2 se aprecian las deficiencias de la estrategia de fuerza bruta “pura”, mientras que las estrategias de Lowe y de comprobación cruzada se comportan de manera similar, no siendo ninguna de ellas claramente superior a la otra en cuanto a su desempeño. Asimismo, la figura 3.4 muestra las limitaciones de SURF a la hora de generar descriptores robustos a cambios moderados en el eje óptico de la cámara.

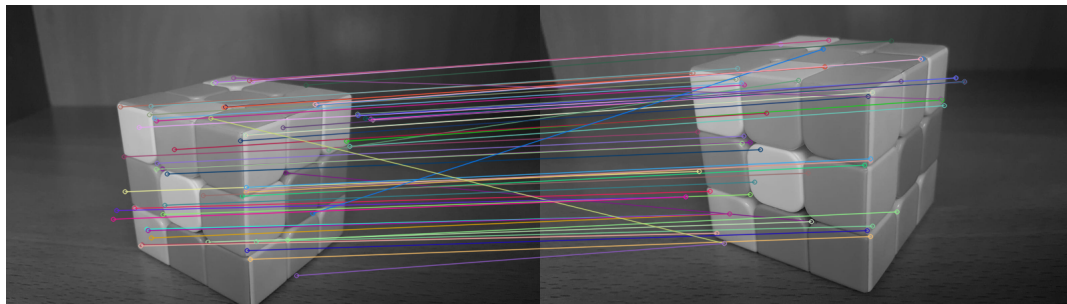


Figura 8.2: Ante perturbaciones en la inclinación del eje óptico de la cámara, únicamente se obtienen buenos emparejamientos cuando estas son del orden de  $\pm 10^\circ$ .

### Extensión afín

Aunque en las figuras 4.3, 4.4 y 4.5 se aprecia un aumento sustancial de la cantidad y calidad de emparejamientos de puntos de interés, el hecho de que la implementación del algoritmo ASIFT no esté integrada en [2] hace difícil su aplicación práctica en el proceso de reconstrucción.

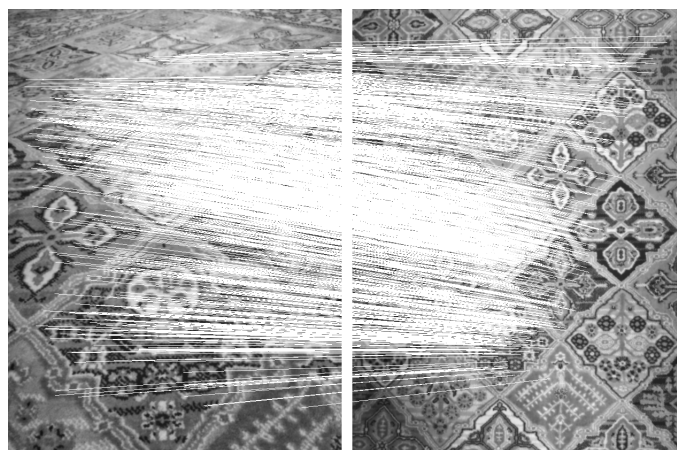


Figura 8.3: ASIFT tiene un comportamiento excelente ante cambios de inclinación arbitrarios cuando la escena está compuesta por un objeto plano.

<sup>1</sup>Tal y como se indica [aquí](#).

<sup>2</sup>[Oficina europea de patentes](#).

### Reconstrucción proyectiva

Los resultados obtenidos en los capítulos 5, 6 y 7 son satisfactorios desde el punto de vista teórico, siendo los únicos problemas encontrados referentes al comportamiento de algunas funcionalidades de [2] y [3] (este último por encontrarse en una fase temprana de desarrollo), que, en algunos casos, como el de la funcionalidad de [2] que implementa el algoritmo descrito en la sección 7.3, tuvieron que ser reprogramadas.

Cabe destacar que, como se sabe que los epipolos son puntos correspondientes, y la precisión en el cálculo de las matrices de cámara depende en gran medida de la precisión en el cálculo de los epipolos, si los epipolos no son aproximadamente correspondientes, la reconstrucción no será fiel. Esta situación se da con frecuencia.

La figura 8.4 muestra dos pares de imágenes idénticas, sobre los cuales se han calculado puntos de interés con SURF, utilizando distintos umbrales Hessianos. Aunque los epipolos geoméricamente son los mismos en ambos pares, su cálculo efectivo es muy distinto en cada caso. Esta situación también se da con mucha frecuencia, pudiendo ser solventada, quizá, con una estrategia tipo RANSAC, ejecutando SURF varias veces con diferentes elecciones de umbral Hessiano.

Estas complicaciones, junto con los efectos de la ambigüedad proyectiva, hacen que las reconstrucciones obtenidas rara vez sean satisfactorias desde el punto de vista práctico.

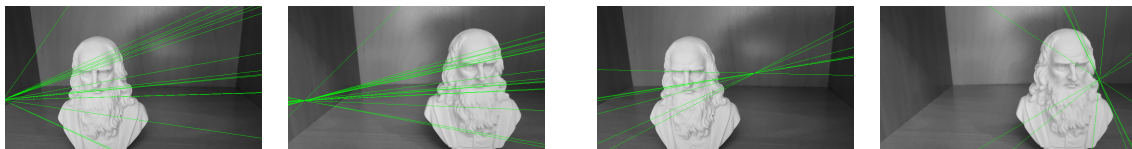


Figura 8.4: Ilustración de algunas de las rectas de los haces de rectas epipolares. Se aprecia que ambos haces, en ambos casos, se cortan, aproximadamente, en el mismo punto de la escena, sin embargo, los puntos de corte son muy distintos en cada par de imágenes.

## 8.2. Conclusiones generales y trabajo futuro

### Conclusiones generales

A lo largo del proyecto se han estudiado los algoritmos para la **detección y descripción de puntos de interés** con más arraigo en la industria a día de hoy, SIFT y SURF, junto con su extensión aún más conocida, ASIFT.

Combinando este estudio con la exposición de las técnicas clásicas de **emparejamiento de descriptores**, ha sido posible establecer procedimientos con los cuales encontrar correspondencias robustas entre imágenes.

Partiendo de esa base, y, tras una breve exposición de la modelización más habitual de las cámaras, la **cámara proyectiva finita**, se han estudiado las **relaciones geométricas entre dos cámaras** que proyectan una misma escena, habiéndose desarrollado procedimientos para, aun desconociendo los parámetros intrínsecos de las cámaras, poder calcularlos (salvo homografías) a partir de las correspondencias de sus retinas.

Una vez conocidos, proyectivamente hablando, los parámetros intrínsecos de las cámaras, se han expuesto algoritmos robustos para el cálculo de una **reconstrucción proyectiva** de la escena.

### Trabajo futuro

Tal y como se ha mencionado, a pesar de haber obtenido resultados satisfactorios, a lo largo del desarrollo del proyecto se han identificado algunos aspectos que merecen en el futuro una dedicación especial

1. Técnicas de **autocalibración** de cámaras para refinar las reconstrucciones proyectivas y conseguir reconstrucciones conformes, o, al menos, afines. Esto sería especialmente prioritario dada la gran distorsión que puede llegar a suponer una homografía, tal y como se aprecia en la figura 7.2.

2. Técnicas para la obtención de **reconstrucciones densas**, es decir, a partir de una nube de puntos como las que se pueden generar con las herramientas de este documento, conseguir una superficie que se ajuste a ella.

Si se decidiera optar por una aproximación calibrada para resolver el problema de las reconstrucciones tridimensionales sería necesario estudiar técnicas de **calibración de cámaras**.

Queda asimismo pendiente explorar la viabilidad de la sustitución de cada uno de los procedimientos expuestos en el documento por **redes neuronales**.

Por otra parte, también puede ser de interés el estudio de algunos operadores posteriores a SURF como BRISK [14] o FREAK [5], que, a pesar de no estar tan implantados en la industria como SURF/SIFT también obtienen buenos resultados y carecen de patentes asociadas.



## Parte IV

## Anexos



## Apéndice A

# Descomposiciones de matrices

### A.1. Descomposición RQ de una matriz $3 \times 3$

En esta sección se presenta un algoritmo para descomponer una matriz cuadrada  $A$  de orden 3 como producto de dos matrices  $R$  y  $Q$ , siendo  $R$  triangular superior y  $Q$  ortogonal, es decir, obtener una igualdad de la forma  $A = RQ$ .

La idea del algoritmo será multiplicar  $A$  por la derecha por una matriz ortogonal  $P$  de manera que  $AP$  sea triangular superior. De esta forma, seleccionando  $R = AP$  y  $Q = P^{-1} = P^T$  se tiene la descomposición deseada. De este modo, la única complicación es encontrar dicha matriz  $P$ . Para ello, se consideran las matrices de rotación con cierto ángulo sobre los ejes

$$R_x := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \quad R_y := \begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix} \quad R_z := \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (A.1)$$

Es claro que al multiplicar la matriz  $A$  por una  $R_x$ , la primera columna del producto es idéntica a la primera columna de  $A$ . Análogamente ocurre al multiplicar por  $R_y$  y  $R_z$ , en cuyo caso la segunda (respectivamente, la tercera) columna de  $A$  coincide con la segunda (respectivamente, la tercera) columna del producto. Al tratarse de rotaciones, todas las matrices de la ecuación (A.1) son ortogonales.

La idea del algoritmo es, seleccionar ciertos ángulos de manera que, multiplicando  $A$  por la izquierda por las matrices de la ecuación A.1 en cierto orden, se obtenga una matriz triangular superior. En concreto, se deben anular los términos  $(2, 1)$ ,  $(3, 1)$  y  $(3, 2)$  del producto.

Antes de hallar los ángulos es conveniente concretar el orden en el que se multiplicarán las rotaciones. Un orden válido es el siguiente:

1. multiplicar por  $R_x$  de manera que se anule el coeficiente  $(3, 2)$ .
2. Multiplicar por  $R_y$  de manera que se anule el coeficiente  $(3, 1)$ . Nótese que el coeficiente  $(3, 2)$  sigue siendo nulo por dejar  $R_y$  la segunda columna invariante.
3. Multiplicar por  $R_z$  de forma que se anule el coeficiente  $(2, 1)$ . Los coeficientes  $(3, 1)$  y  $(3, 2)$  seguirán siendo nulos, por ser la primera columna del producto una combinación lineal de las dos primeras columnas de  $AR_xR_y$ , teniendo ambas dos columnas un coeficiente nulo en su tercera fila.

Una vez hallado el orden correcto de multiplicación, es necesario hallar los ángulos, que deben cumplir, si se denota  $A = (a_{ij})_{i,j}$ , las siguientes ecuaciones

$$\begin{aligned} a_{32} \cos \alpha + a_{33} \sin \alpha &= 0 \\ a_{31} \cos \beta - a_{33} \sin \beta &= 0 \\ a'_{21} \cos \gamma + a'_{22} \sin \gamma &= 0 \end{aligned} \quad (A.2)$$

Donde  $a'_{21}$  y  $a'_{22}$  son los coeficientes  $(2, 1)$  y  $(2, 2)$  del producto  $AR_xR_y$ .

Unas soluciones a estas ecuaciones son

$$\cos \alpha = \frac{a_{33}}{\sqrt{a_{33}^2 + a_{32}^2}} \quad \cos \beta = \frac{a_{33}}{\sqrt{a_{33}^2 + a_{31}^2}} \quad \cos \gamma = \frac{a'_{22}}{\sqrt{a_{22}^2 + a_{21}^2}} \quad (\text{A.3})$$

De este modo,  $\mathbf{A} = (\mathbf{A}\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z)(\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z)^\top$ , siendo la primera matriz del producto triangular superior, y la segunda ortogonal, tal y como se requería.

## A.2. Descomposición en valores singulares (SVD)

**Teorema A.2.1.** *Dada una matriz  $\mathbf{A}$   $m \times n$ , entonces, existen dos matrices ortogonales  $\mathbf{U}$   $m \times m$  y  $\mathbf{V}$   $n \times n$  tales que  $\mathbf{U}^\top \mathbf{A} \mathbf{V} = \mathbf{D}$ , siendo  $\mathbf{D}$  una matriz diagonal no negativa  $m \times n$ . Los elementos en la diagonal de  $\mathbf{D}$  se llaman **valores singulares** de  $\mathbf{A}$ .*

*Demostración.* Si  $\mathbf{A} = \mathbf{0}$  no hay nada que demostrar. En otro caso, se considera el vector unitario  $\mathbf{v}_1$  que maximiza  $\|\mathbf{A}\mathbf{v}_1\|^2$ , este está bien definido por ser la aplicación  $\mathbf{v} \mapsto \|\mathbf{A}\mathbf{v}\|^2$  continua sobre la esfera unidad, que es un conjunto compacto.

Además, el vector  $\mathbf{v}_1$  es un punto crítico de la función lagrangiana asociada al problema de minimizar la aplicación  $\mathbf{v} \mapsto \|\mathbf{A}\mathbf{v}\|^2$ . Esta función lagrangiana es

$$L(\mathbf{v}, \lambda) = \|\mathbf{A}\mathbf{v}\|^2 + \lambda(1 - \|\mathbf{v}\|^2) \quad (\text{A.4})$$

Cuyo gradiente respecto de  $\mathbf{v}$  es

$$\nabla_{\mathbf{v}} L(\mathbf{v}, \lambda) = 2(\mathbf{v}^\top \mathbf{A}^\top \mathbf{A} - \lambda \mathbf{v}^\top) \quad (\text{A.5})$$

Luego, por ser  $\mathbf{v}_1$  punto crítico, se cumple

$$\mathbf{v}_1^\top \mathbf{A}^\top \mathbf{A} = \lambda \mathbf{v}_1^\top \quad (\text{A.6})$$

Sabiendo esto, y, denotando  $\mathbf{u}_1 := \frac{\mathbf{A}\mathbf{v}_1}{\|\mathbf{A}\mathbf{v}_1\|}$ , es posible demostrar que la aplicación lineal asociada a  $\mathbf{A}$  envía el espacio ortogonal a  $\mathbf{v}_1$ , que se denota por  $\langle \mathbf{v}_1 \rangle^\perp$  al espacio ortogonal a  $\mathbf{u}_1$ ,  $\langle \mathbf{u}_1 \rangle^\perp$ .

En efecto, dado  $\mathbf{v}$  un vector ortogonal a  $\mathbf{v}_1$ , basta con considerar el producto escalar

$$(\mathbf{A}\mathbf{v})^\top \mathbf{u}_1 = \mathbf{v}^\top \mathbf{A}^\top \mathbf{u}_1 = \frac{1}{\|\mathbf{A}\mathbf{v}_1\|} \mathbf{v}^\top \mathbf{A}^\top \mathbf{A} \mathbf{v}_1 = \frac{1}{\|\mathbf{A}\mathbf{v}_1\|} \lambda \mathbf{v}^\top \mathbf{v}_1 = 0 \quad (\text{A.7})$$

Luego, considerando las bases ortonormales  $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ ,  $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ , y denotando  $\sigma_1 := \|\mathbf{A}\mathbf{v}_1\|$ , se tiene, usando las relaciones de ortogonalidad, el siguiente resultado

$$\mathbf{U}_1^\top \mathbf{A} \mathbf{V}_1 := \begin{pmatrix} \mathbf{u}_1^\top \\ \vdots \\ \mathbf{u}_m^\top \end{pmatrix} \mathbf{A} (\mathbf{v}_1 \quad \dots \quad \mathbf{v}_n) = \begin{pmatrix} \mathbf{u}_1^\top \mathbf{A} \mathbf{v}_1 & \mathbf{u}_1^\top \mathbf{A} \mathbf{v}_2 & \dots & \mathbf{u}_1^\top \mathbf{A} \mathbf{v}_n \\ \mathbf{u}_2^\top \mathbf{A} \mathbf{v}_1 & \mathbf{u}_2^\top \mathbf{A} \mathbf{v}_2 & \dots & \mathbf{u}_2^\top \mathbf{A} \mathbf{v}_n \\ \vdots & \vdots & \dots & \vdots \end{pmatrix} = \begin{pmatrix} \sigma_1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{A}_1 \end{pmatrix} \quad (\text{A.8})$$

Si se repite este mismo proceso con  $\mathbf{A}_1$  usando matrices  $\mathbf{U}_2$ ,  $\mathbf{V}_2$  y se consideran las matrices

$$\overline{\mathbf{U}}_2 = \begin{pmatrix} \mathbf{I}_1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{U}_2 \end{pmatrix} \quad \overline{\mathbf{V}}_2 = \begin{pmatrix} \mathbf{I}_1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{V}_2 \end{pmatrix} \quad (\text{A.9})$$

multiplicando por bloques se tiene que

$$\overline{\mathbf{U}}_2^\top \mathbf{U}_1^\top \mathbf{A} \mathbf{V}_1 \overline{\mathbf{V}}_2 = \begin{pmatrix} \mathbf{I}_1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{U}_2 \end{pmatrix} \begin{pmatrix} \sigma_1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{A}_1 \end{pmatrix} \begin{pmatrix} \mathbf{I}_1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{V}_2 \end{pmatrix} = \begin{pmatrix} \sigma_1 & \mathbf{0}^\top \\ \mathbf{0} & \mathbf{U}_2^\top \mathbf{A}_1 \mathbf{V}_2 \end{pmatrix} \quad (\text{A.10})$$

Lo cual demuestra que el proceso realizado con  $\mathbf{A}_1$  se puede concatenar con el realizado sobre  $\mathbf{A}$ , luego concatenando tantos procesos como sea posible se obtiene el resultado final, ya que las matrices ortogonales son cerradas respecto del producto.  $\blacksquare$

En el capítulo 2 (páginas 59–70) de [25] pueden encontrarse algoritmos para el cálculo de descomposiciones en valores singulares de matrices en general.

## Apéndice B

# Geometría del punto unidad

Se considera el plano proyectivo complejo canónico  $\mathbb{P}^2$ , al cual se le supone dotado de una estructura conforme inducida por una recta  $l_\infty$  y dos puntos  $\Omega_\infty := \{i, j\} \subset l_\infty$ . En el capítulo 5 se expone un procedimiento con el cual es posible construir una referencia afín de  $\mathbb{P}^2$  con ejes ortogonales. En este capítulo se extiende dicho procedimiento, dándose un método geométrico para calcular un punto unidad alternativo de manera que la referencia resultante es conforme.

### B.1. Caracterización del punto unidad

**Teorema B.1.1.** *Sean  $x_0, x_1$  conjugados respecto a  $\Omega_\infty$  y  $x_2 \notin l_\infty$ , entonces,  $\{x_0, x_1, x_2; e\}$  es una referencia conforme respecto de  $\Omega_\infty$  si y solo si las rectas  $x_2x_0, x_2x_1$ , y sus respectivas paralelas por  $e$  forman un cuadrado.*

*Demostración.* Toda la demostración se apoya en los cálculos reflejados en la figura B.1. Además, es conveniente recordar que las rectas  $x_2x_0, x_2x_1$  y sus paralelas por  $e$  forman un cuadrado si y solo si sus diagonales  $l_1$  y  $l_2$  son perpendiculares.

Por una parte, si  $\{x_0, x_1, x_2; e\}$  es una referencia conforme, por definición, la matriz de  $\Omega_\infty$  es la identidad  $2 \times 2$ , y las rectas  $l_1$  y  $l_2$  tienen por direcciones a los puntos  $(1 : -1 : 0)$  y  $(1 : 1 : 0)$  respectivamente, siendo dichas direcciones trivialmente ortogonales.

Recíprocamente, si se escoge  $e$  de manera que las rectas  $x_2x_0, x_2x_1$  y sus paralelas por  $e$  forman un cuadrado, es claro que  $e$  no está sobre las rectas que conforman el triángulo cuyos vértices son  $x_0, x_1$  y  $x_2$ , luego  $\{x_0, x_1, x_2; e\}$  es una referencia. Además, por ser  $x_0$  y  $x_1$  conjugados respecto de  $\Omega_\infty$ , la matriz de  $\Omega_\infty$  es diagonal. Por último, como por hipótesis  $l_1$  y  $l_2$  son ortogonales, se debe cumplir

$$(1 \quad 1) \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = a - b = 0 \iff a = b \quad (\text{B.1})$$

Por ende,  $\{x_0, x_1, x_2; e\}$  es una referencia conforme. ■

### B.2. Procedimiento de construcción

Del teorema B.1.1 se concluye que para construir un punto unidad adecuado para una referencia afín con ejes ortogonales, basta con construir un cuadrado, siendo dos de sus lados los ejes de la referencia. A continuación se demuestra que se puede construir un cuadrado usando únicamente las primitivas de la geometría conforme, es decir, las circunferencias.

**Teorema B.2.1.** *Dado un plano  $\mathbb{P}^2$  con estructura conforme, un punto  $c$  y dos rectas perpendiculares por él, si se traza una circunferencia con centro  $c$  y se consideran las intersecciones de dicha circunferencia con las perpendiculares, entonces, las perpendiculares junto con las rectas paralelas a ellas que pasan por las intersecciones consideradas, forman un cuadrado.*

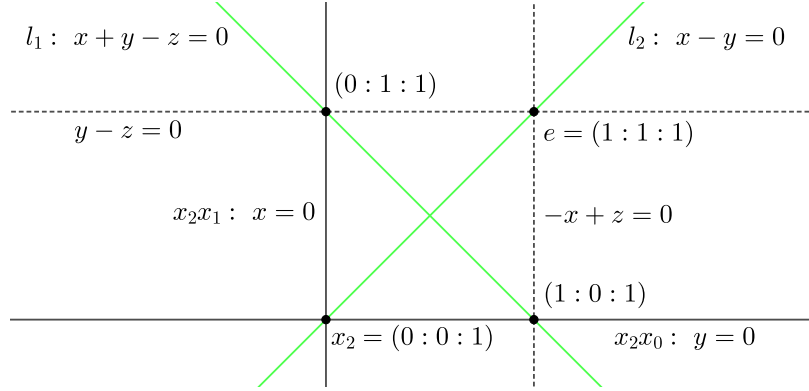


Figura B.1: Ilustración de algunos cálculos necesarios para la demostración del teorema B.1.1.

*Demostración.* Tomando una referencia conforme adaptada  $\{x_0, x_1, c; e\}$ , los círculos con centro  $c$  son cónicas que pasan por  $\{i, j\}$  cuyo polo respecto de  $l_\infty$  es  $c$ .

En primer lugar, es necesario ver cómo son las matrices de las circunferencias que pasan por  $c$ . Partiendo de una matriz simétrica genérica, el hecho de que el polo de  $l_\infty$  sea  $c$  implica que se debe cumplir

$$\begin{pmatrix} a & h & g \\ h & b & f \\ g & f & c \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \iff g = 0, f = 0, c = 1 \quad (\text{B.2})$$

Además, de la condición de pasar por  $i = (1 : i : 0)$  y  $j = (1 : -i : 0)$  surgen las ecuaciones

$$\begin{aligned} a - 2ih - b &= 0 \\ a + 2ih - b &= 0 \end{aligned} \quad (\text{B.3})$$

De donde se deduce que  $a = b$  y que  $h = 0$ . Siendo la matriz (normalizada) de una circunferencia que pasa por  $c$  de la forma

$$\begin{pmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{B.4})$$

La figura B.2 ilustra los cálculos de las intersecciones de una circunferencia genérica con los ejes, así como de las rectas paralelas a los ejes por estas intersecciones y sus diagonales  $l_1$  y  $l_2$ , que tienen por direcciones a  $\left(\sqrt{\frac{-1}{a}} : \sqrt{\frac{-1}{a}} : 0\right)$  y  $\left(-\sqrt{\frac{-1}{a}} : \sqrt{\frac{-1}{a}} : 0\right)$  respectivamente, luego son trivialmente ortogonales. ■

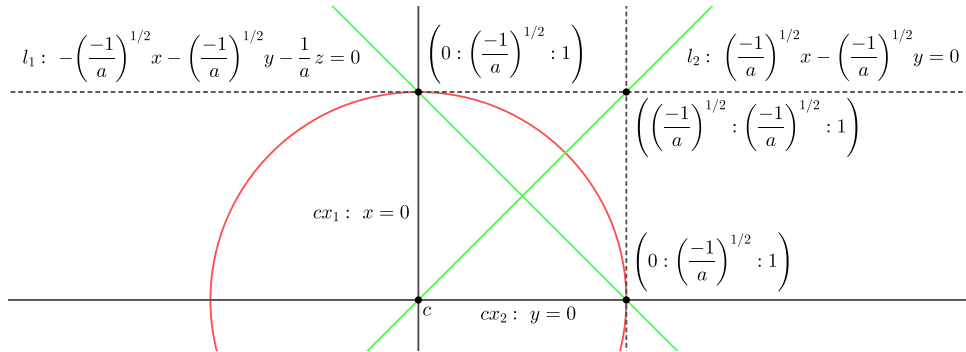


Figura B.2: Ilustración de algunos cálculos necesarios para la demostración del teorema B.2.1.

## Apéndice C

# Código fuente

Todos los programas realizados a modo de material complementario de este documento han sido desarrollados en lenguaje [Python](#), y están disponibles en el repositorio público de [GitHub](#) “[projective-reconstructions](#)” [11] bajo la licencia [AGPLv3](#).

En dicho repositorio se puede encontrar una pequeña [guía de usuario](#), donde se explica cómo utilizar cada uno de los programas. En este anexo se explica brevemente la estructura de algunas de las funciones que componen el programa principal, relacionándolas con los contenidos teóricos del proyecto.

El programa principal, el cual recibe como entrada dos imágenes, de las cuales obtiene su reconstrucción proyectiva, se corresponde con el archivo “[reco.py](#)”. Dicho archivo contiene un conjunto de funciones que son usadas a lo largo de el proceso de reconstrucción.

### **compute\_keypoints**

Esta función toma dos imágenes y un número (umbral Hessiano), calculando sus puntos de interés (junto con sus descriptores correspondientes) usando SURF. Finalmente, genera dos nuevas imágenes representando los puntos de interés detectados sobre las imágenes originales.

### **compute\_correspondences**

Calcula los emparejamientos de descriptores usando la regla de Lowe, genera una representación gráfica de los mismos y devuelve dos listas del mismo tamaño, siendo los elementos  $i$ -ésimos de cada una de ellas, las coordenadas de dos puntos correspondientes.

### **compute\_fundamental\_matrix**

Calcula la matriz fundamental con un algoritmo basado en RANSAC a partir de las listas devueltas por la función de cálculo de emparejamientos. Además, genera una representación gráfica del haz de rectas epipolares de cada imagen, calcula los epipolos y elimina de las listas de puntos correspondientes a aquellas parejas que no están en el soporte dado por RANSAC.

### **compute\_cameras**

Calcula las matrices de las cámaras a partir de la matriz fundamental y uno de los epipolos, según la ecuación (6.25), escogiendo aleatoriamente los parámetros libres.

### **triangulate**

Calcula una triangulación de los puntos de la escena usando el procedimiento expuesto en la sección 7.3.





# Bibliografía

- [1] Geogebra. <https://www.geogebra.org/>.
- [2] OpenCV: (Open Source Computer Vision Library). <https://opencv.org/>.
- [3] Point Processing Tool Kit (PPTK). <https://github.com/heremaps/pptk>.
- [4] SageMath. <http://www.sagemath.org/es/>.
- [5] Alahi, Alexandre, Raphael Ortiz y Pierre Vandergheynst: *FREAK: Fast Retina Keypoint*. páginas 510–517, Julio 2012.
- [6] Bay, Herbert, Tinne Tuytelaars y Luc Van Gool: *SURF: Speeded Up Robust Features*. Computer Vision and Image Understanding, 110:346–359, Junio 2008.
- [7] Ehsan, Shoaib, Adrian F. Clark, Naveed ur Rehman y Klaus D. McDonald-Maier: *Integral Images: Efficient Algorithms for Their Computation and Storage in Resource-Constrained Embedded Vision Systems*. Sensors, 15:16804–16830, Julio 2015, ISSN 1424-8220.
- [8] Fernando, José F. y J. Manuel Gamboa: *Geometría Lineal: Espacios Afines y Projectivos*. Sanz y Torres, 2018, ISBN 978-84-16466-49-8.
- [9] Fischler, Martin A. y Robert C. Bolles: *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Communications of the ACM, 24:381–395, Junio 1981.
- [10] Forsyth, David A. y Jean Ponce: *Computer Vision, a modern approach*. Pearson Prentice Hall, 2003, ISBN 0-13-191193-7.
- [11] García Álvaro: *projective-reconstructions*. <https://github.com/alvarogatenorio/projective-reconstructions>.
- [12] Hartley, Richard y Andrew Zisserman: *Multiple View Geometry in computer vision*. Cambridge University Press, 2003, ISBN 0-521-54051-8.
- [13] Hartley, Richard I.: *Euclidean reconstruction from uncalibrated views*. Second Europe-US Workshop on Invariance, Ponta Delgada, Azores, páginas 187–202, Octubre 1993.
- [14] Leutenegger, Stefan, Margarita Chli y Roland Siegwart: *BRISK: Binary Robust invariant scalable keypoints*. páginas 2548–2555, Noviembre 2011.
- [15] Longuet-Higgins, H.C.: *A computer algorithm for reconstructing a scene from two projections*. Nature, 293:133–135, Septiembre 1981.
- [16] Lowe, David G.: *Distinctive Image Features from Scale-Invariant Keypoints, cascade filtering approach*. International Journal of Computer Vision, 60:91–110, Enero 2004.
- [17] Luong, Quan Tuan y Oliver D. Faugeras: *The Fundamental Matrix: Theory, Algorithms, and Stability Analysis*. International Journal of Computer Vision, páginas 43–75, Enero 1996.
- [18] López, Clara I. y Álvaro García: *Geometría Lineal*. <https://github.com/alvarogatenorio/Geometria-Lineal/blob/master/geoli.pdf>.

- [19] Merino, Luis y Evangelina Santos: *Álgebra Lineal con métodos elementales*. Paraninfo, 2014, ISBN 978-84-9732-481-6.
- [20] Moons, Theo, Luc Van Gool y Maarten Vergauwen: *3D Reconstruction from Multiple Images: Part 1, Principles*. Foundations and Trends in Computer Graphics and Vision, 4:287–404, Enero 2009.
- [21] Olsson, Carl: *Computer Vision*. Lund Institute of Technology, Lund University, 2019. <http://www.maths.lth.se/matematiklth/personal/calle/datorseende19/index.html>.
- [22] Oyallon, Edouard y Julien Rabin: *An Analysis of the SURF Method*. Image Processing On Line, 5:176–218, Julio 2015.
- [23] Pajares, Gonzalo y Jesús M. De la Cruz: *Visión por Computador: Imágenes digitales y aplicaciones*. RA-MA, 2001, ISBN 978-8478974726.
- [24] Pang, Yanwei, Wei Li, Yuan Yuan y Jing Pan: *Fully affine invariant SURF for image matching*. Elsevier Neurocomputing, 85:6–10, Diciembre 2011.
- [25] Press, William H., Saul A. Teukolsky, William T. Vetterling y Brian P. Flannery: *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992, ISBN 0-521-43108-5.
- [26] Rodríguez, Mariano, Julie Delon y Jean Michel Morel: *Fast Affine Invariant Image Matching*. Image Processing On Line, 8:251–281, Septiembre 2018.
- [27] Valdés, Antonio: *Geometría Computacional*.
- [28] Valdés, Antonio y Diego Chicharro: *Geometría Proyectiva*.
- [29] Yu, Guoshen y Jean Michel Morel: *ASIFT: An Algorithm for Fully Affine Invariant Comparison*. Image Processing On Line, 1:11–38, Febrero 2011.

# Índice general

- autocalibración, 59
- blobs, 16
- cámara
  - afín, 34
  - calibrada, 47
  - proyectiva finita, 33
- cónica
  - del absoluto, 34
- calibración
  - de cámaras, 60
- centro
  - de cámara, 33
  - de cámara, 27
- comprobación
  - cruzada, 26
- convolución, 11
- ecuación
  - de Joachimstahl, 34
- eje
  - principal, 27, 34
- epipolos, 40
- filtros de Haar, 21
- forma canónica, 44
- función
  - cóncava en un punto, 15
  - convexa en un punto, 15
- imagen, 11
  - convolucionada, 12
  - integral, 13
- matriz
  - asociada a un par de cámaras, 42
  - fundamental, 42
  - intrínseca, 37
- parámetros
  - extrínsecos, 38
- plano
  - del infinito, 34
  - imagen, 33
  - epipolar, 39
- proyección
  - cónica, 33
- punto
  - de interés, 7, 11
  - principal, 34
- puntos
  - correspondientes, 42
- reconstrucción, 49
  - densa, 60
- recta
  - base, 39
  - epipolar, 39
- red
  - neuronal, 60
- referencia
  - canónica de la retina, 34
- retina, 33
- soporte, 49
- supresión no máxima, 19
- valores
  - singulares, 64
- visión
  - estereoscópica, 27